

FIG. 1

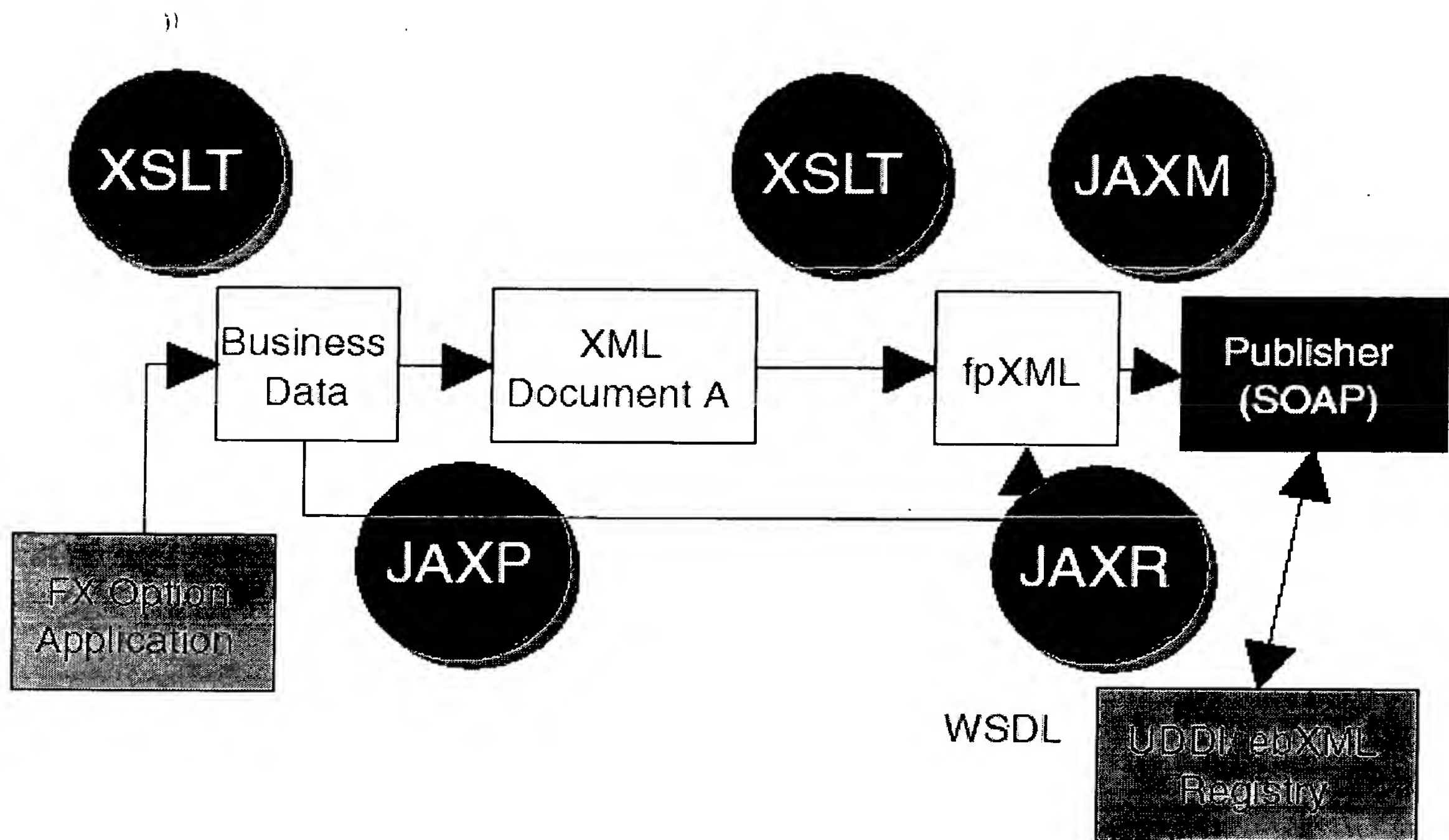


FIG. 2

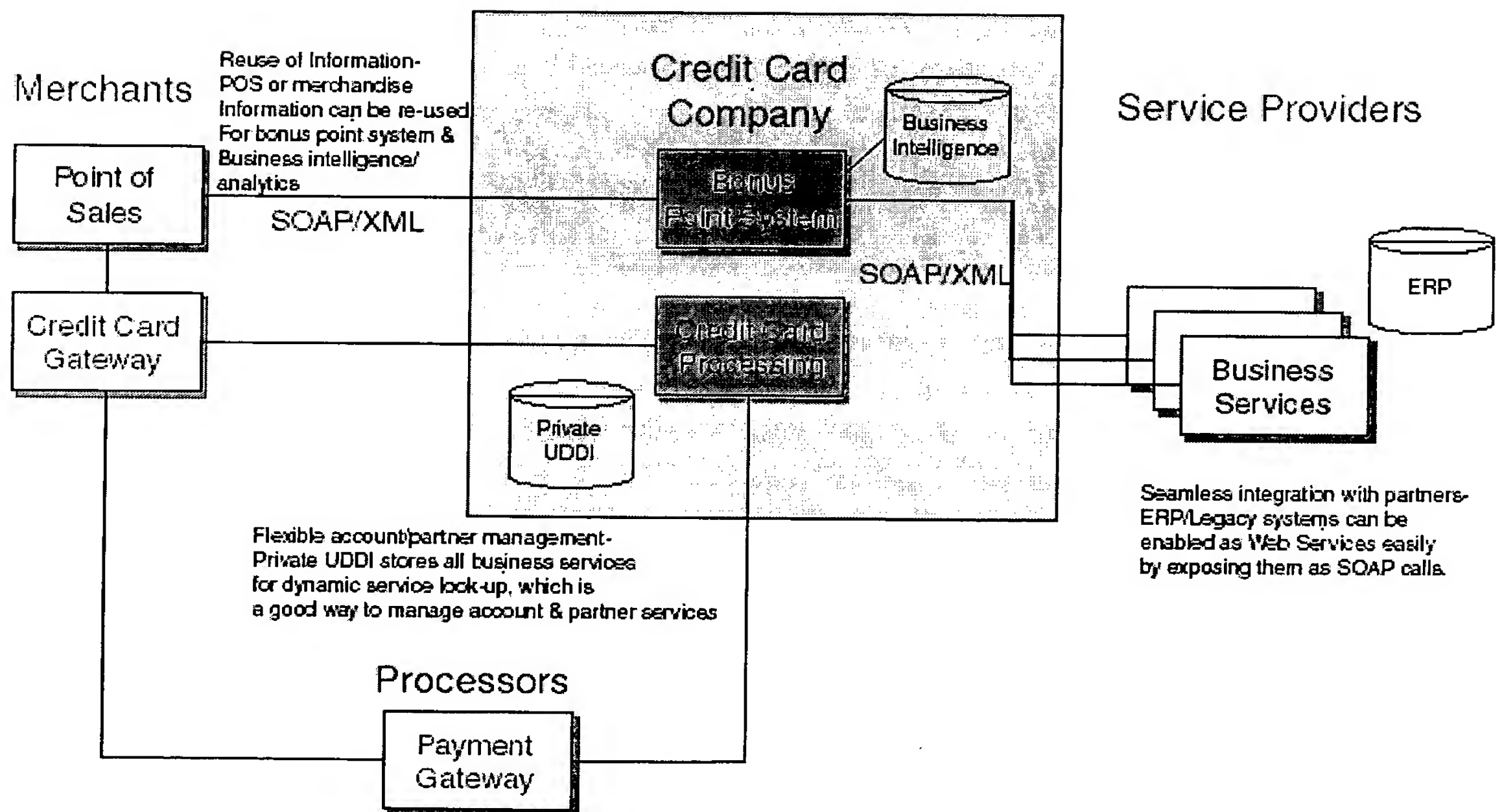


FIG. 3

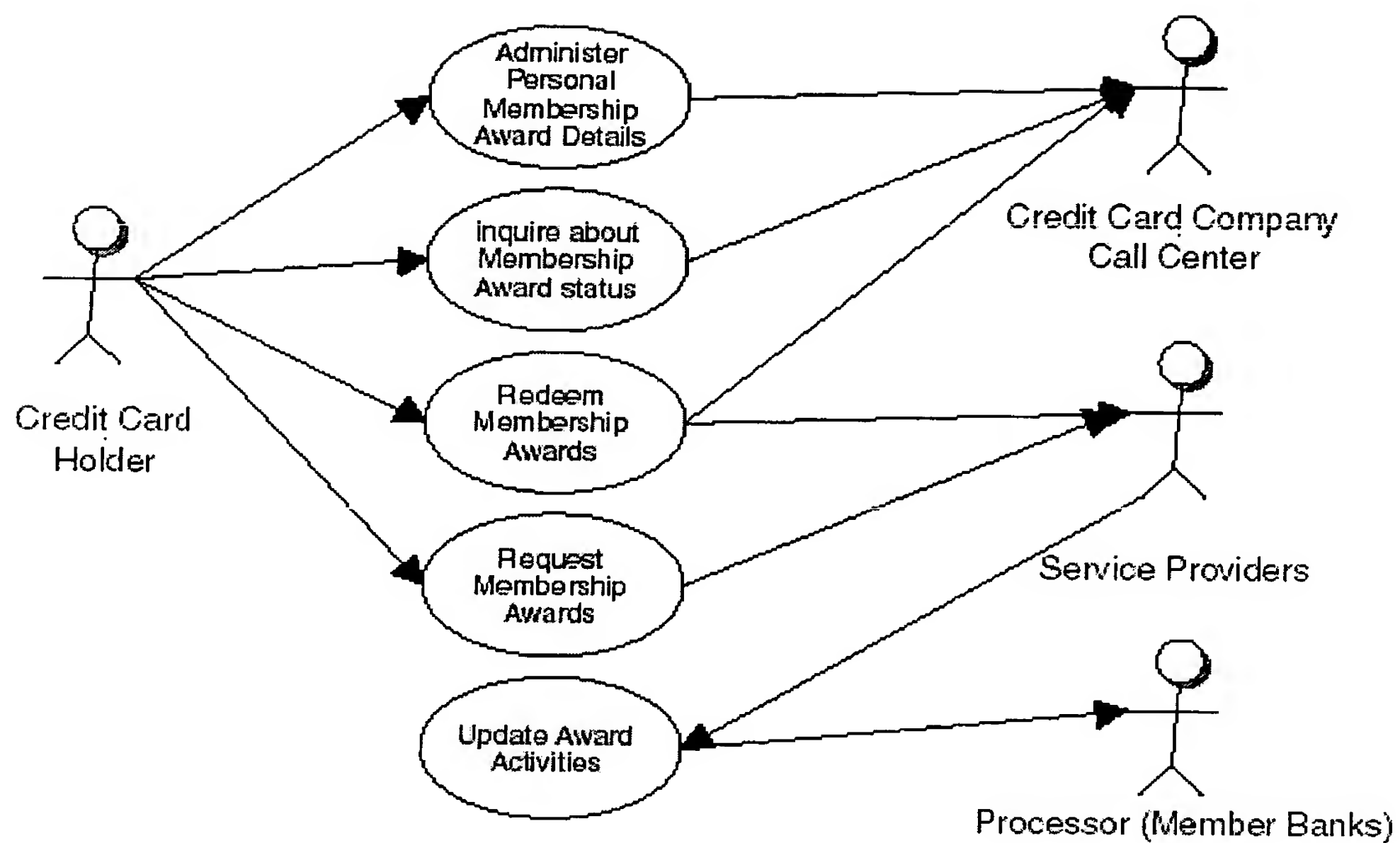


FIG. 4

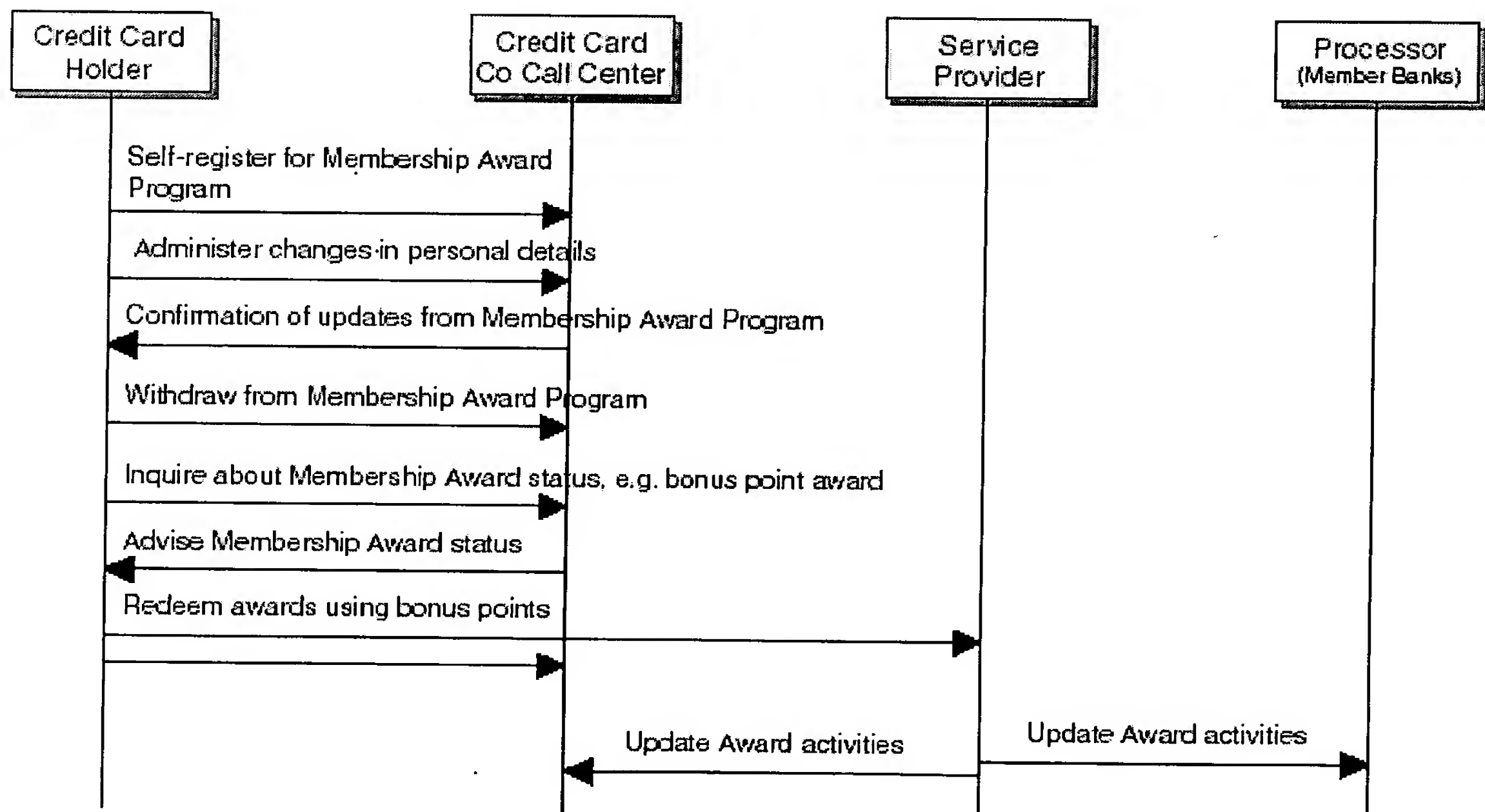


FIG. 5

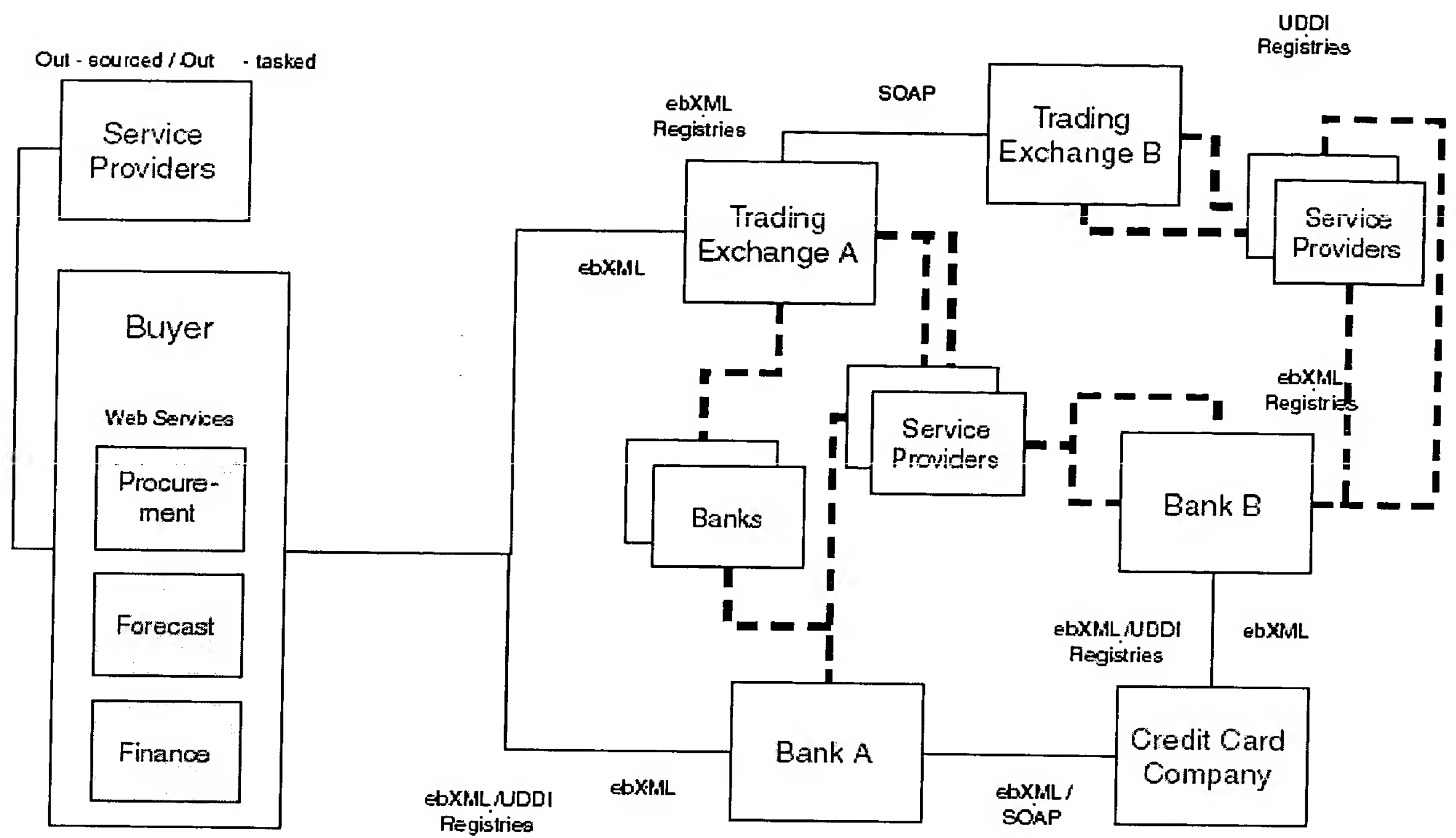


FIG. 6

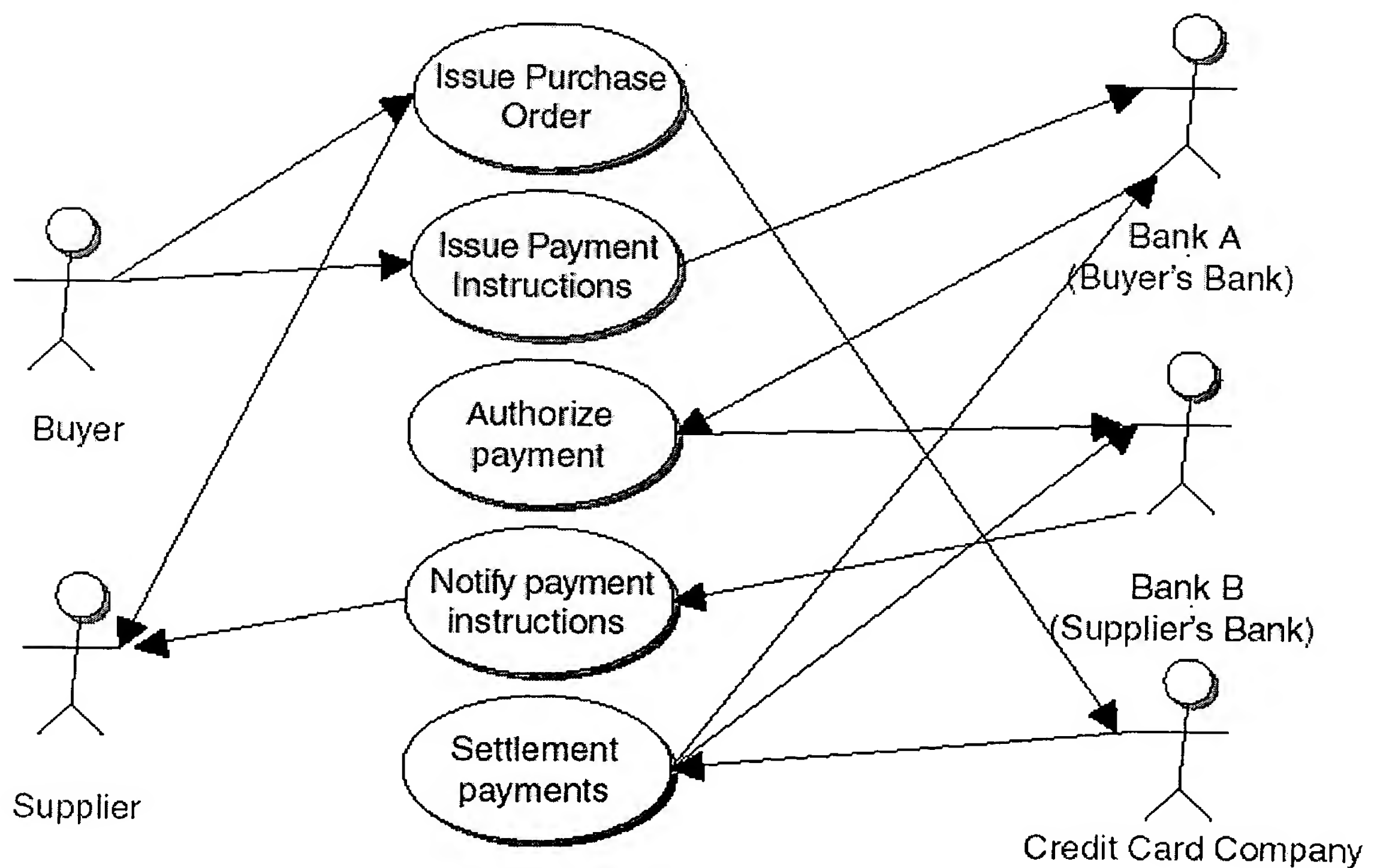


FIG. 7



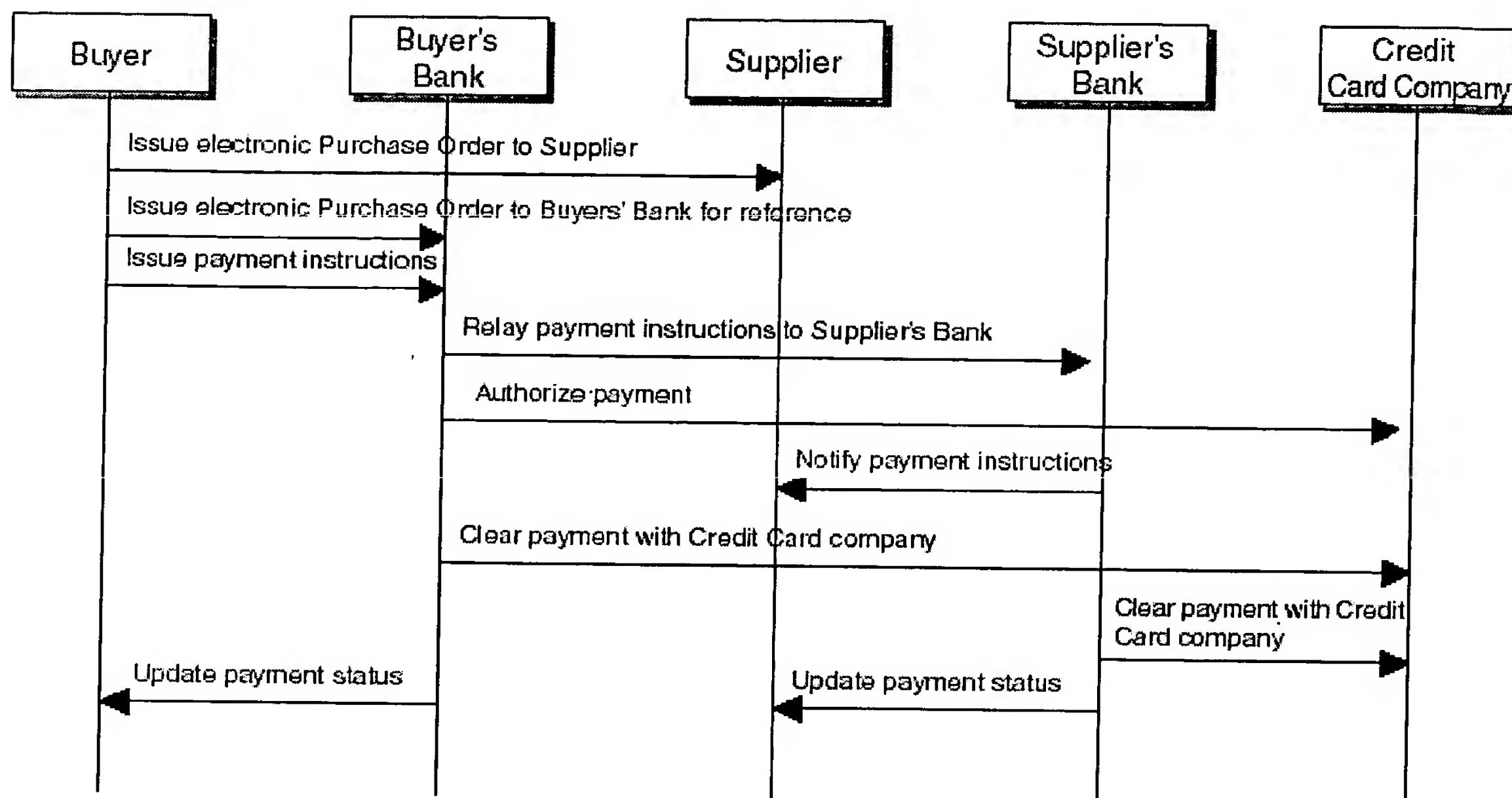


FIG. 8

Examples of  
These Protocols/Services:

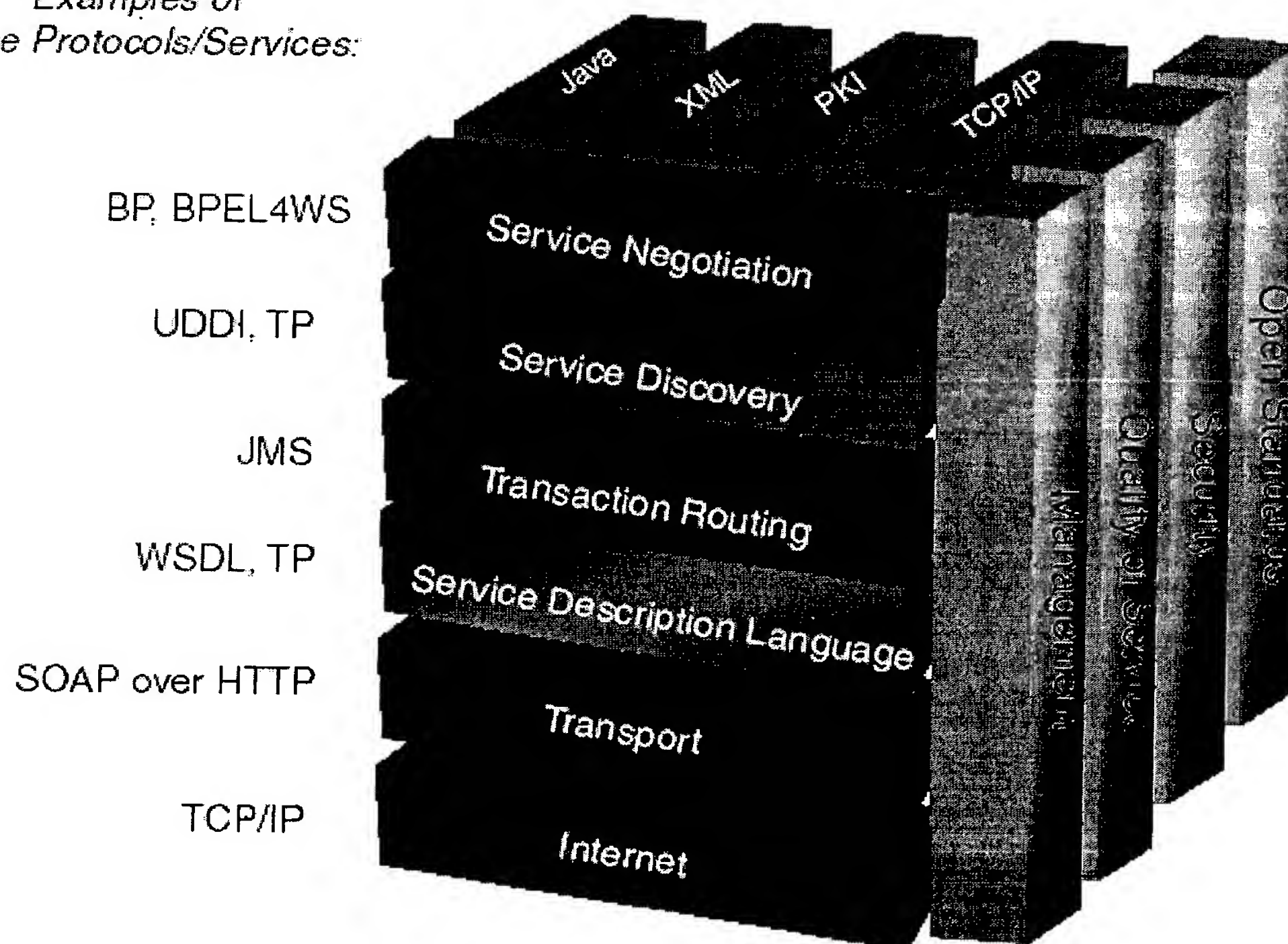


FIG. 9

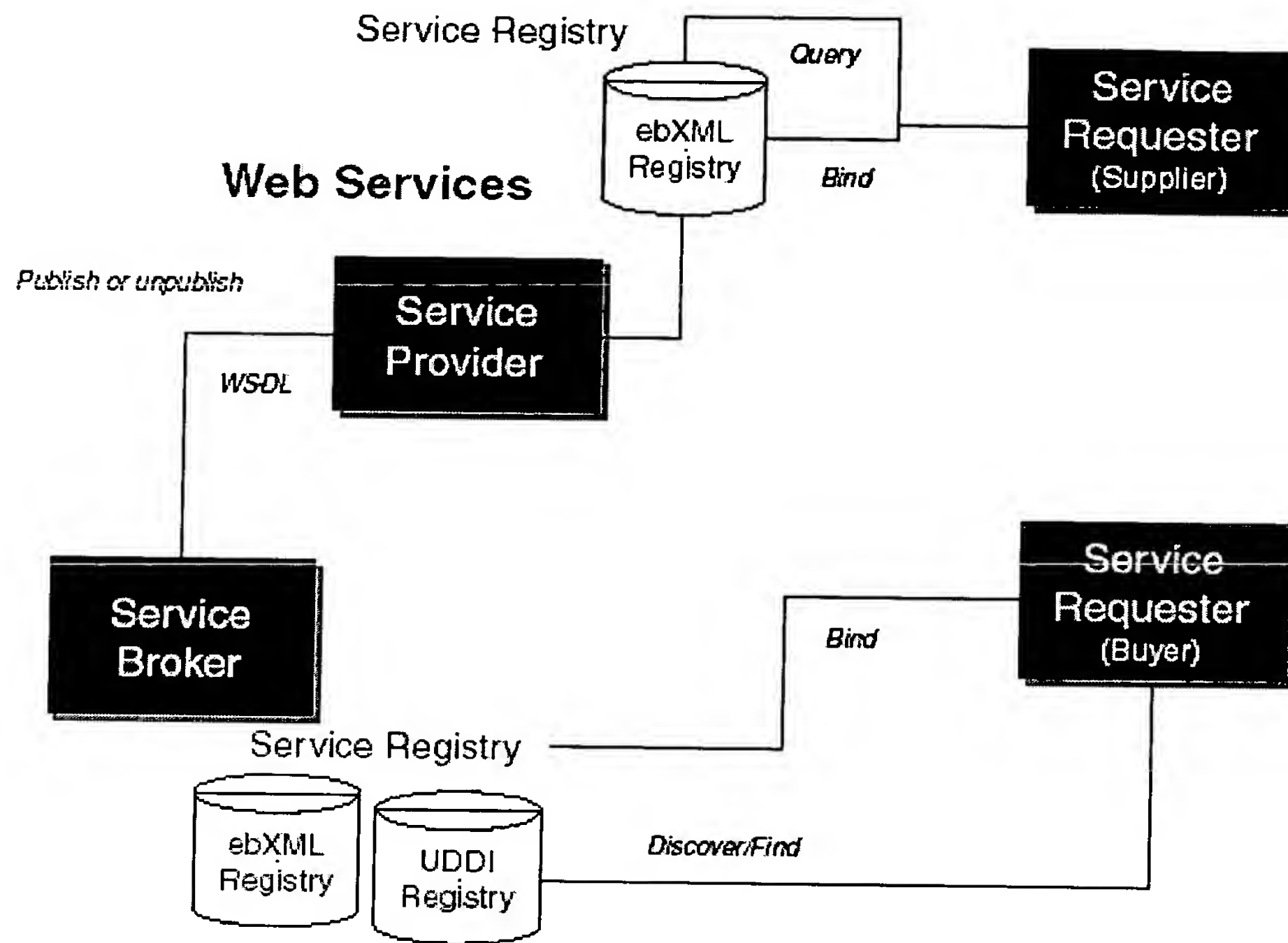


FIG. 10

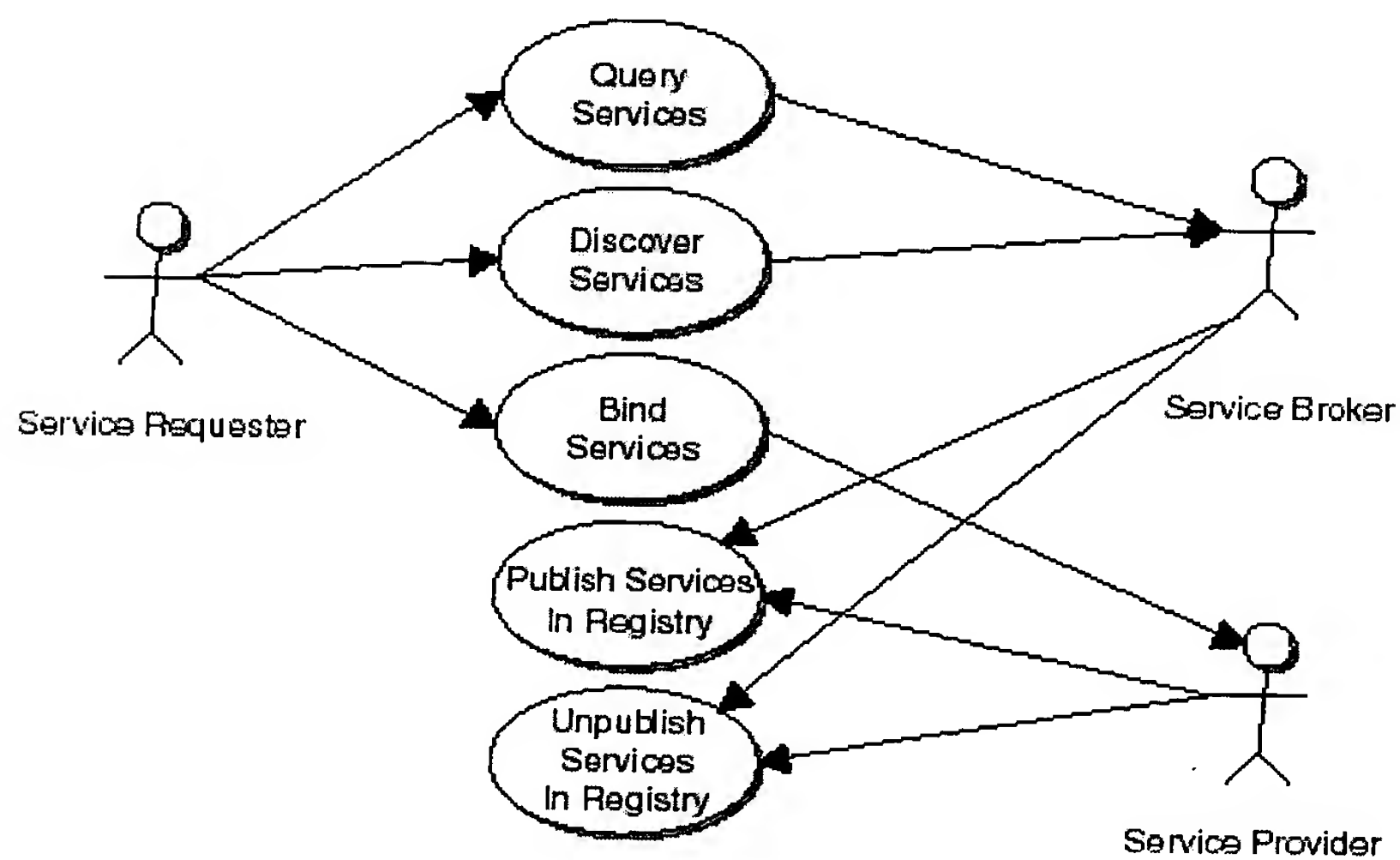


FIG. 11

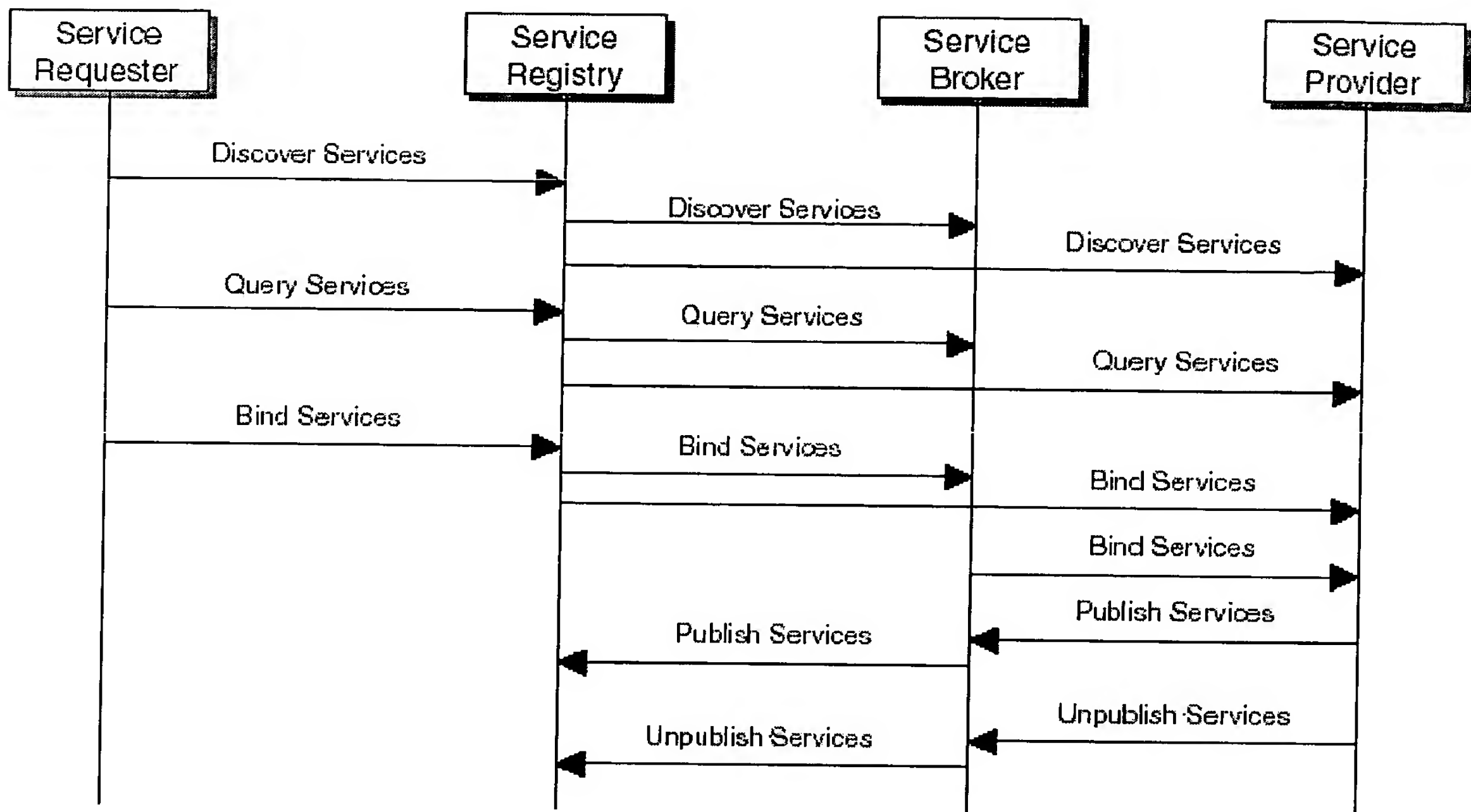


FIG. 12

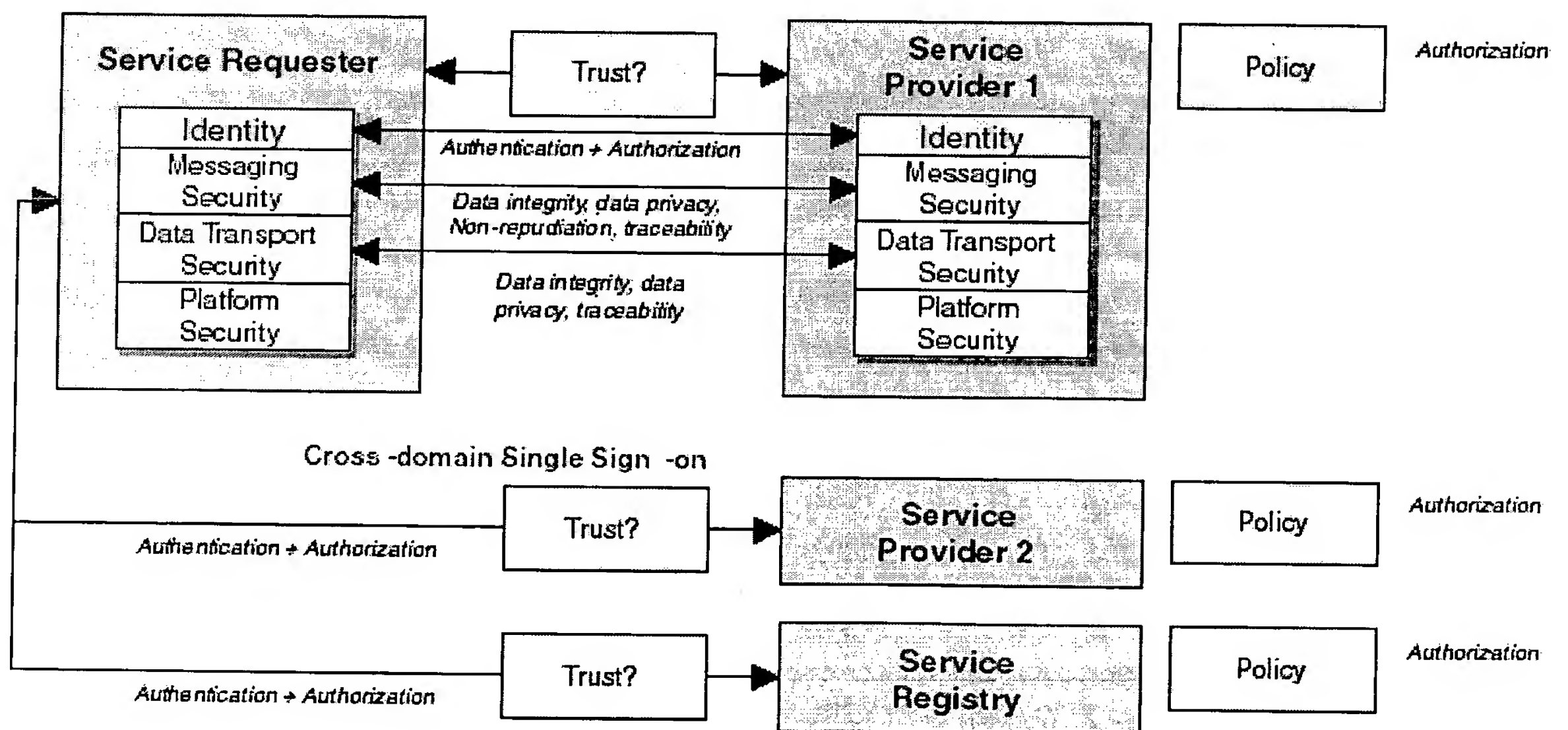


FIG. 13

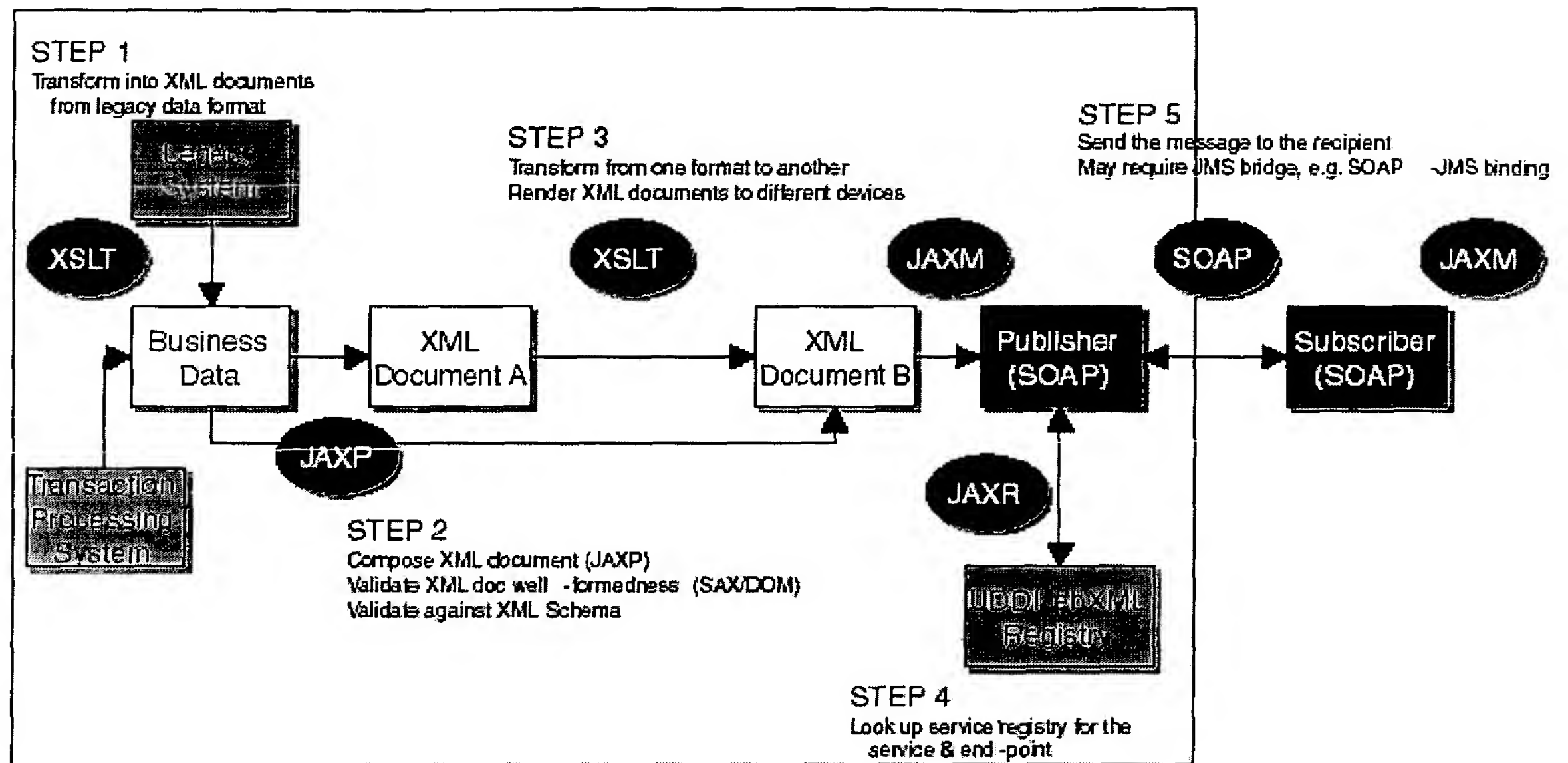


FIG. 14

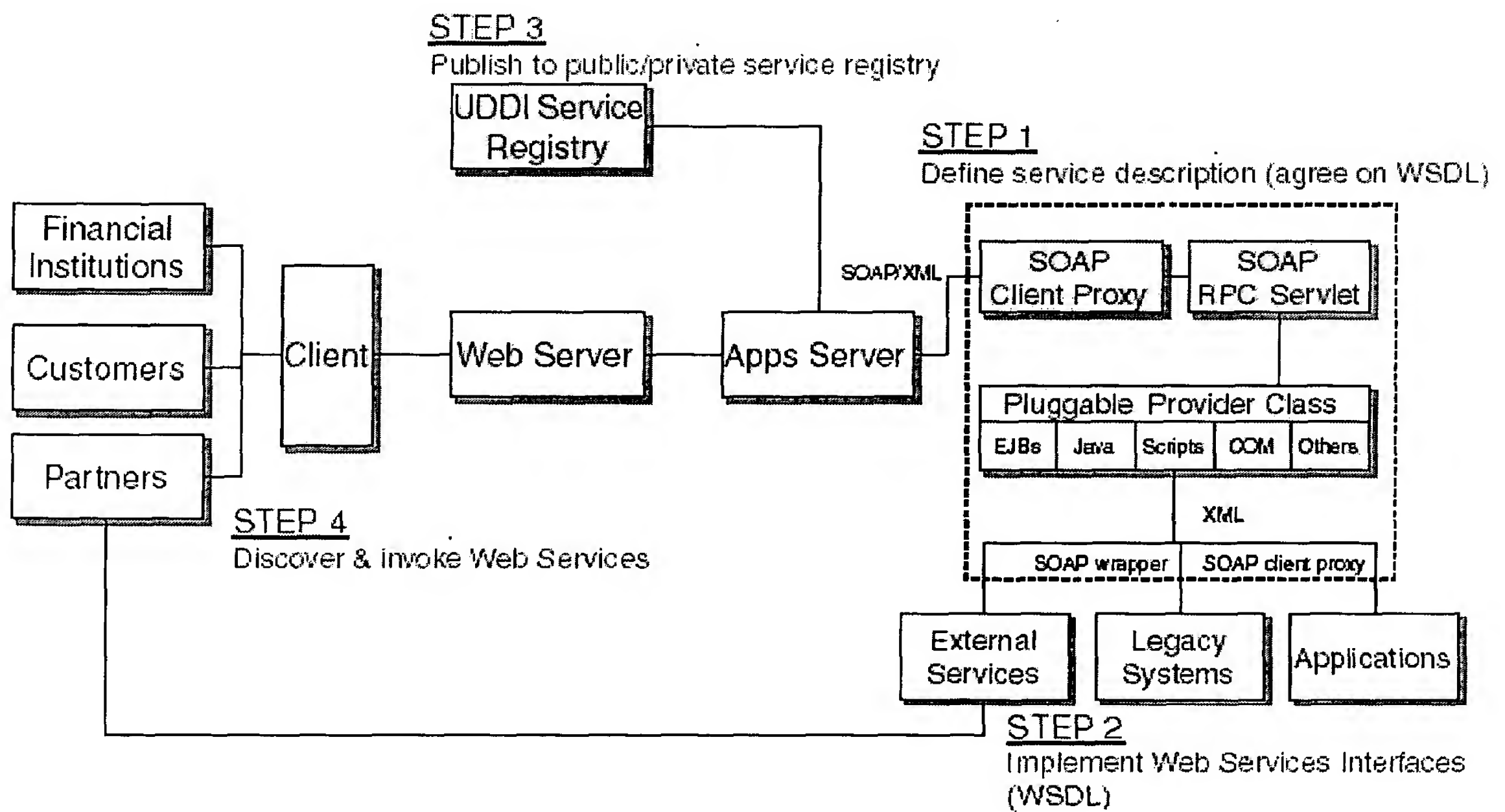


FIG. 15

Out-tasked / Out-sourced

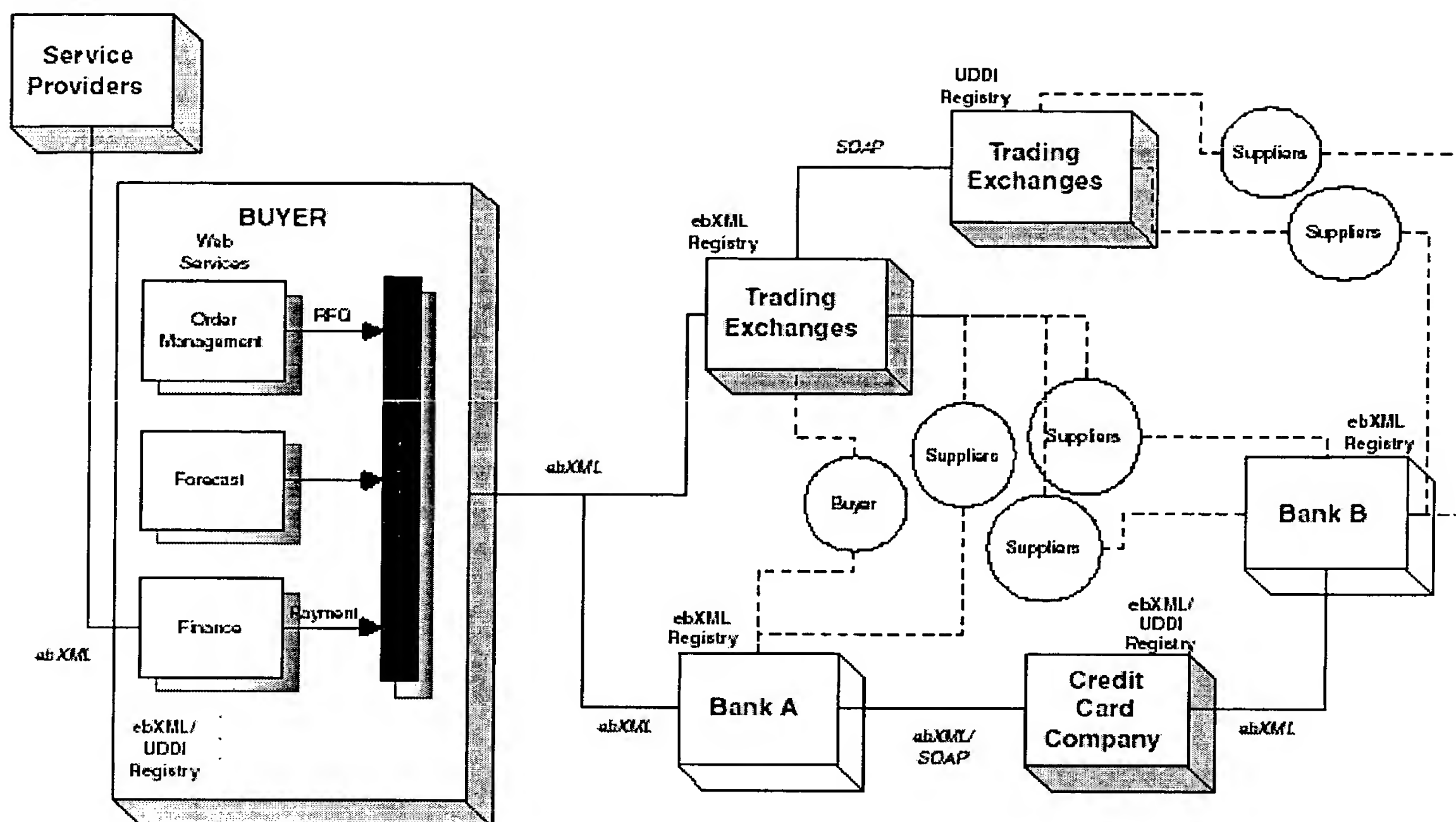


FIG. 16

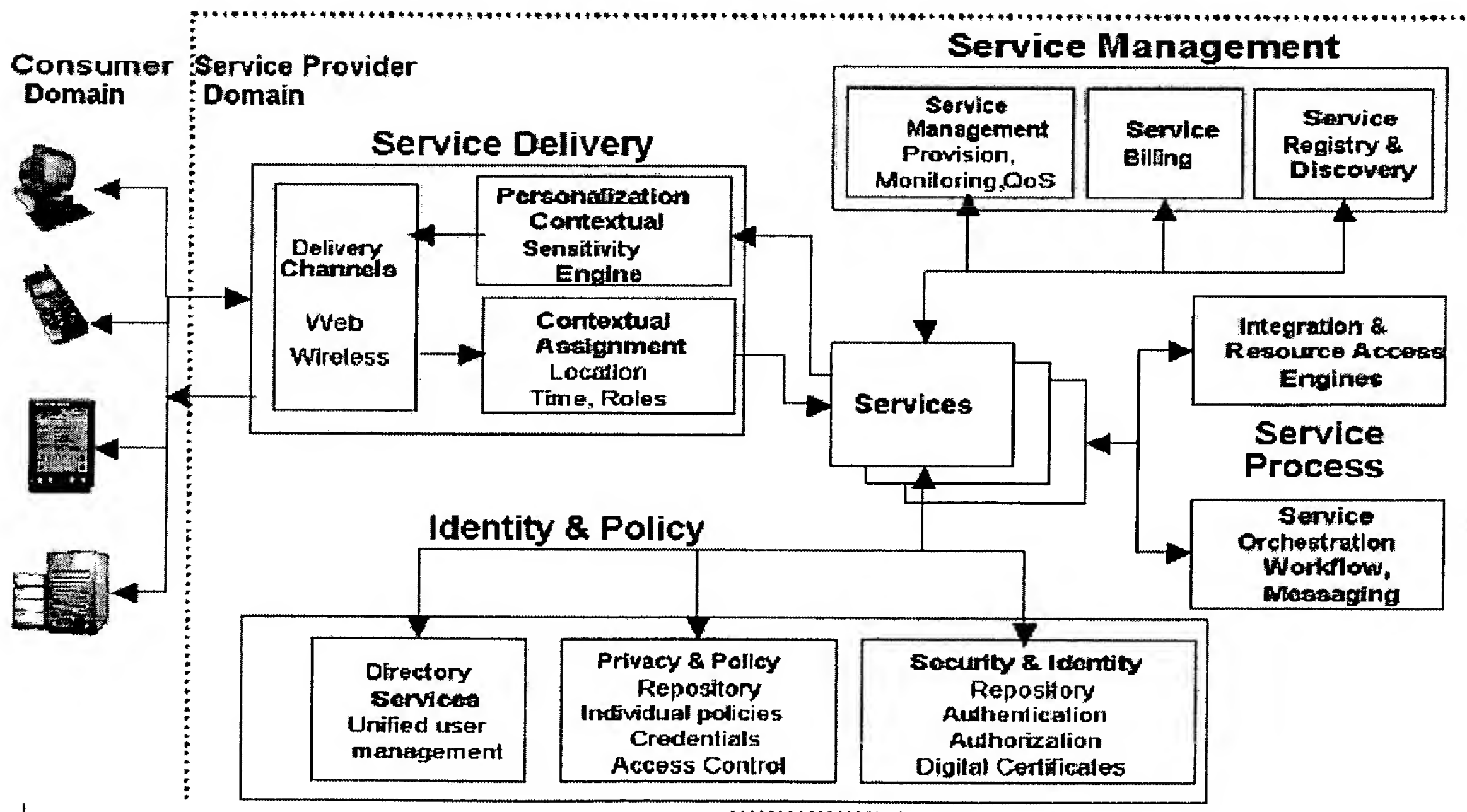


FIG. 17



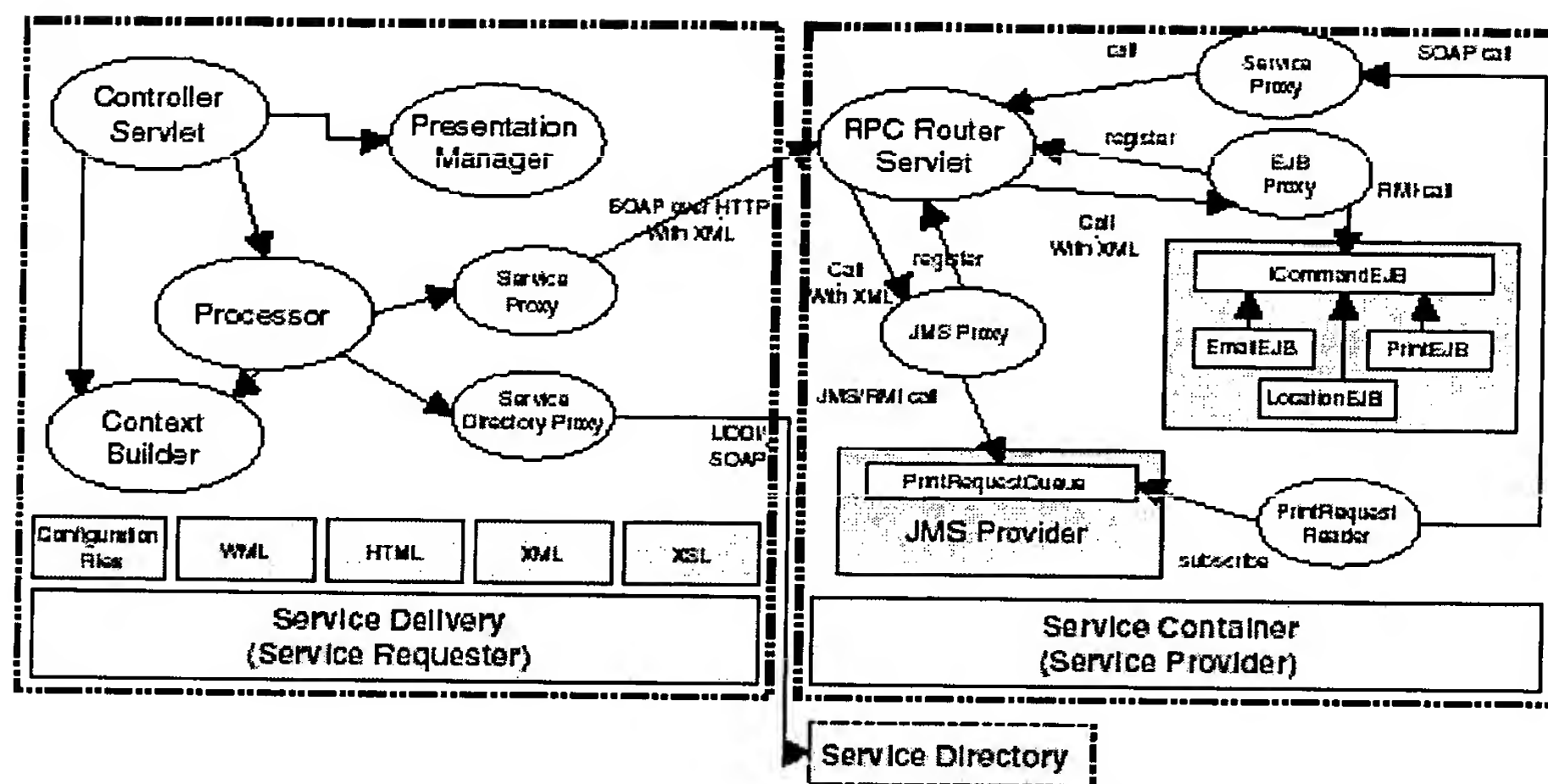


FIG. 18

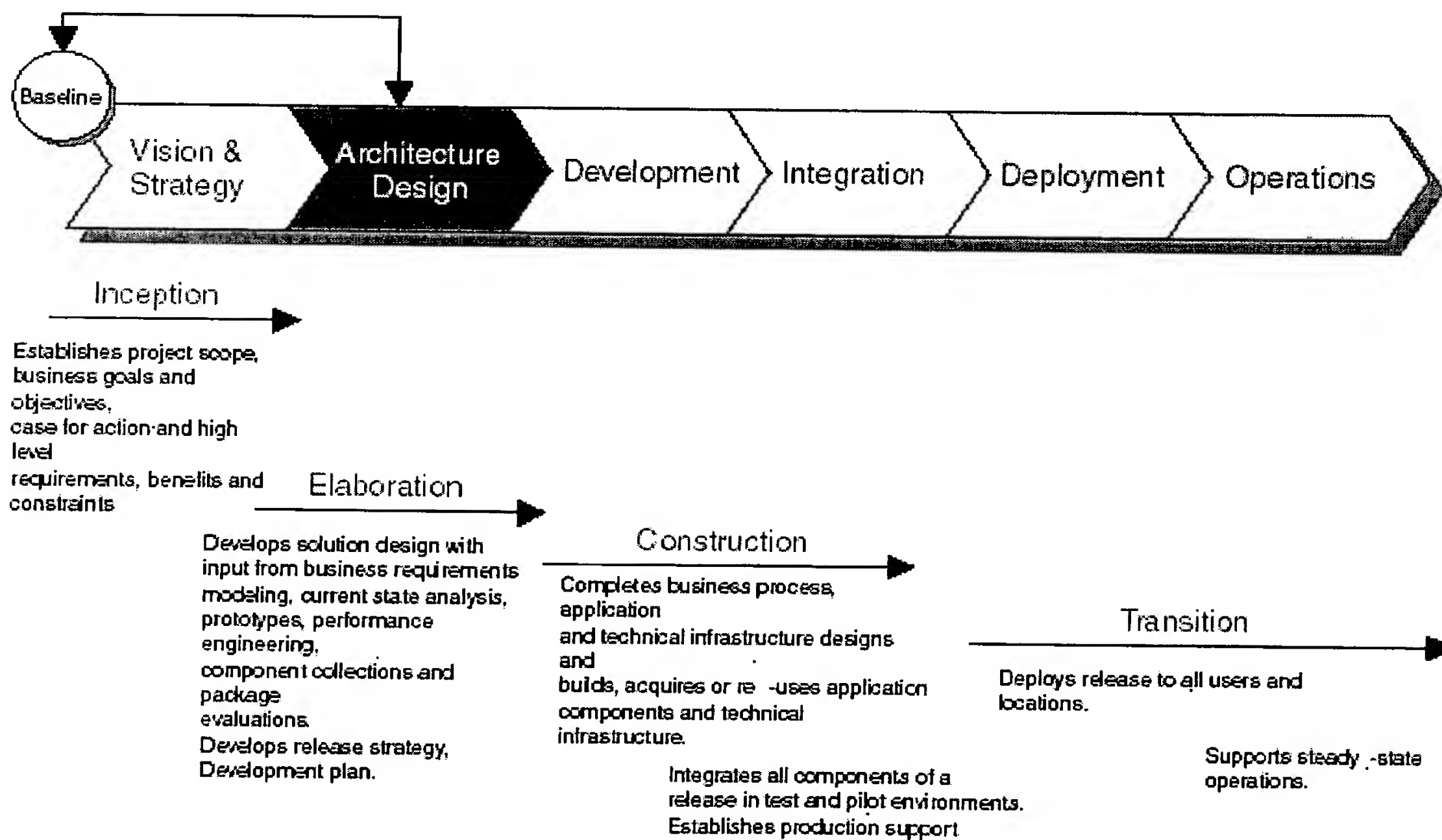


FIG. 19

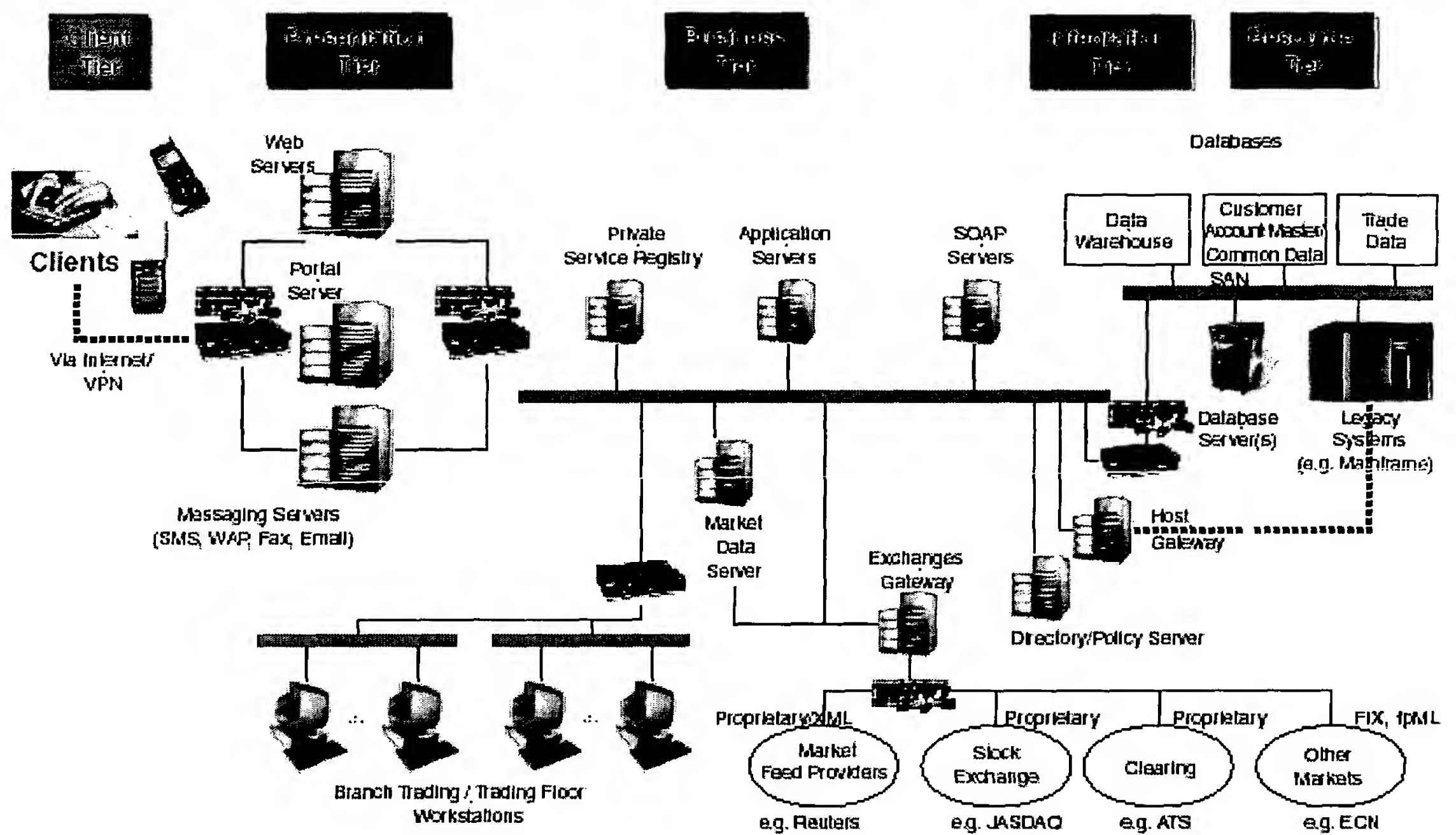


FIG. 20

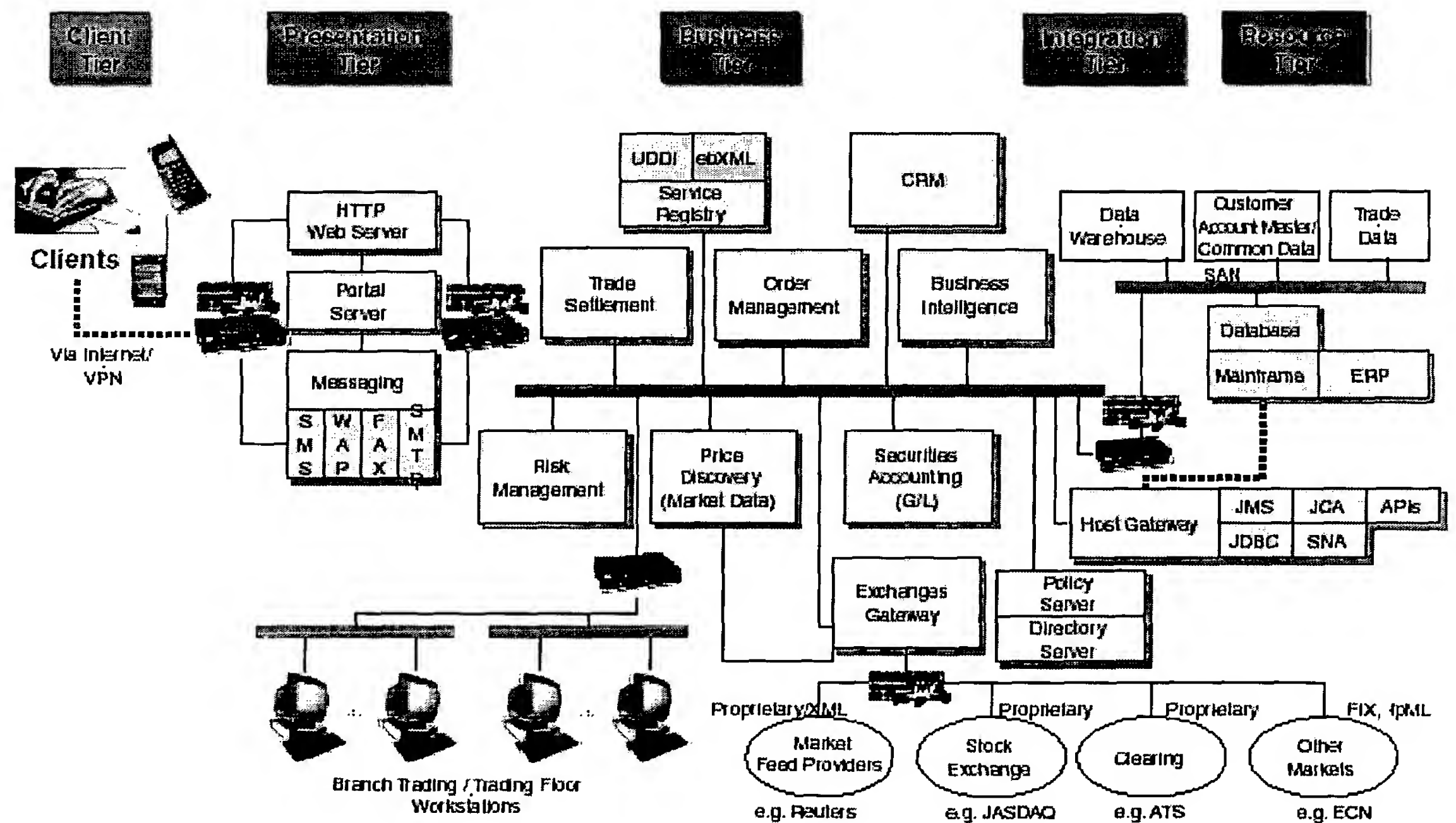


FIG. 21



Tiers/ Platform Layer	Client Tier	Presentation Tier	Business Tier	Integration Tier	Resource Tier
Application Platform Layer			Order management Trade settlement Risk management Price discovery Securities accounting CRM Business Intelligence	Service Registry	ERP systems Policy Server Directory Server
Virtual Platform Layer		J2EE		SOAP ebXML	Policy Server Directory Server
Upper Platform Layer	Client Browser	Messaging Servers Web Server Portal Server	Application Server		Database Server Policy Server Directory Server
Lower Platform Layer	PDA WAP phone	Solaris OE	Solaris OE	Solaris OE	Policy Server Directory Server
Hardware Platform Layer	PDA WAP phone	Sparc Unix			Mainframe Storage devices/SAN

FIG. 22

"ilities"	Client Tier	Presentation Tier	Business Tier	Integration Tier	Resource Tier
Performance, throughput, and scalability		HTTP-based load balancing for SOAP servlet SOAP/XML cache	Vertical scaling Horizontal scaling	HTTP-based load balancer for Service Registry SOAP/XML cache	Federated Directory Server
Reliability and availability	Reliable and clustered hardware platform	Reliable and clustered hardware platform Clustered messaging servers	Reliable and clustered hardware platform Clustered Application Server	Clustered Service Registry	Master-slave Directory Server for HA Parallel database server Standby database server Reliable and clustered hardware platform
Security	HTTPS VPN gateway	HTTPS VPN gateway	HTTPS	XML security (e.g., DSIG, WS-security)	XML security standards (e.g., SAML, XACML) Trusted Solaris OE
Manageability	System management tools	System management tools	System management tools	System management tools	System management tools
Flexibility		Decoupling presentation from business (e.g., XML for data, HTML for presentation)		Update URL end-point in Service Registry without re-binding run-time (re-compilation)	
Reusability			SOAP-enabled business services	SOAP-enabled business services	SOAP-enabled business services

FIG. 23

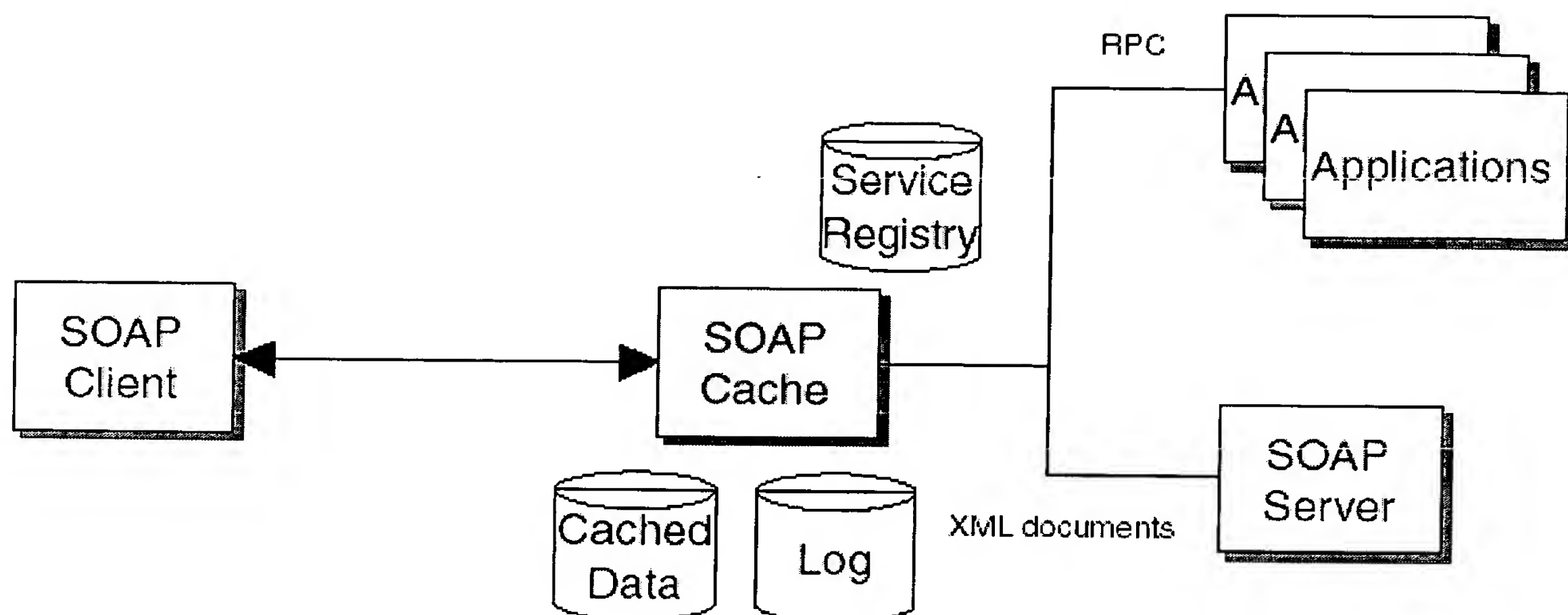


FIG. 24

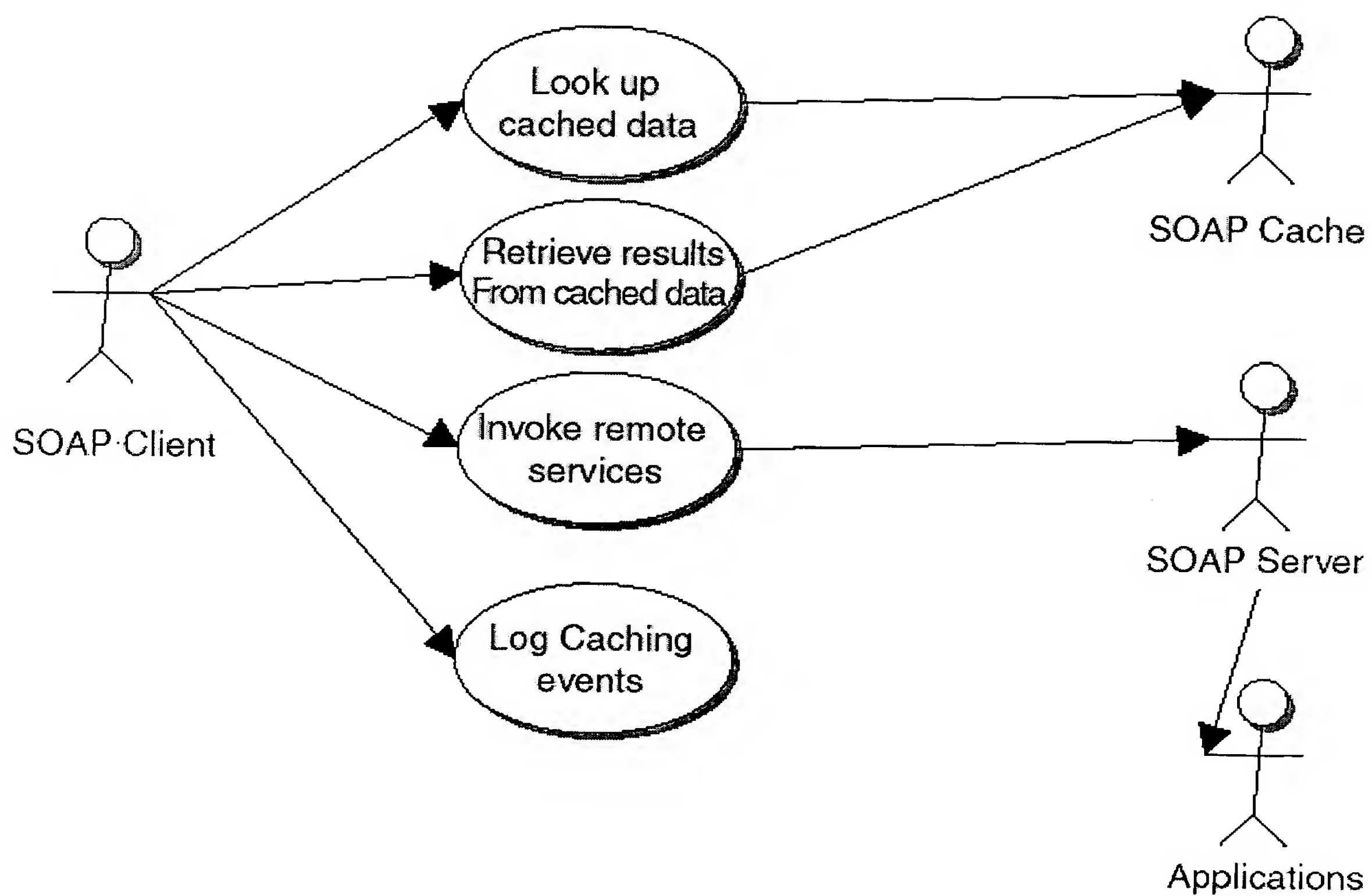


FIG. 25

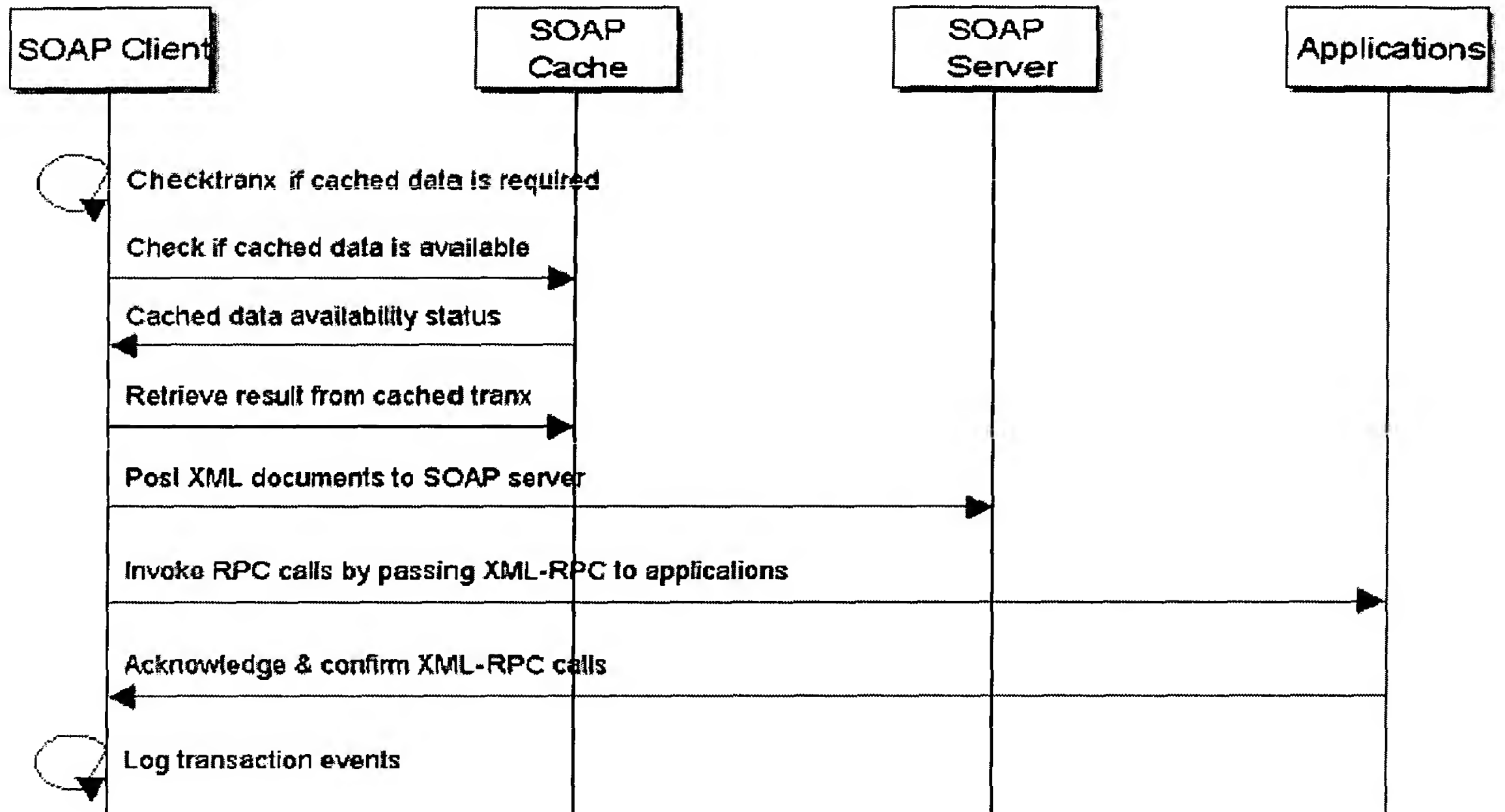


FIG. 26

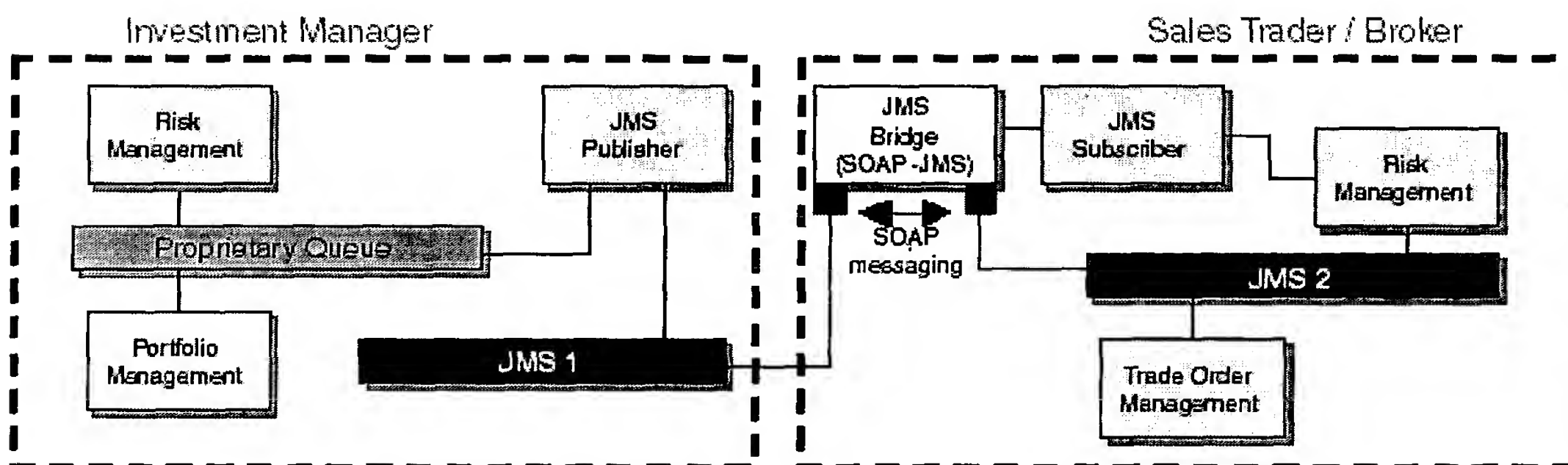


FIG. 27

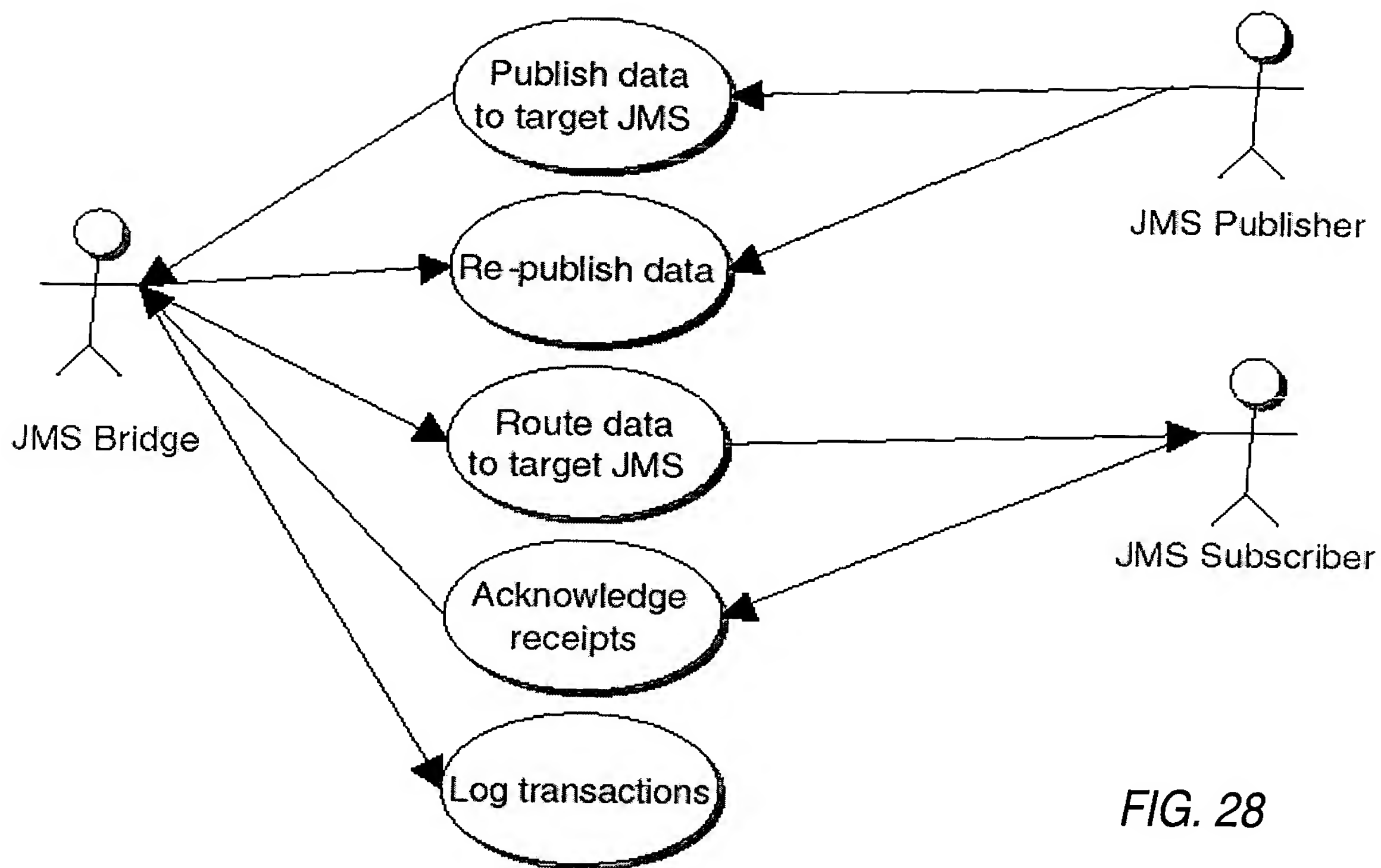


FIG. 28

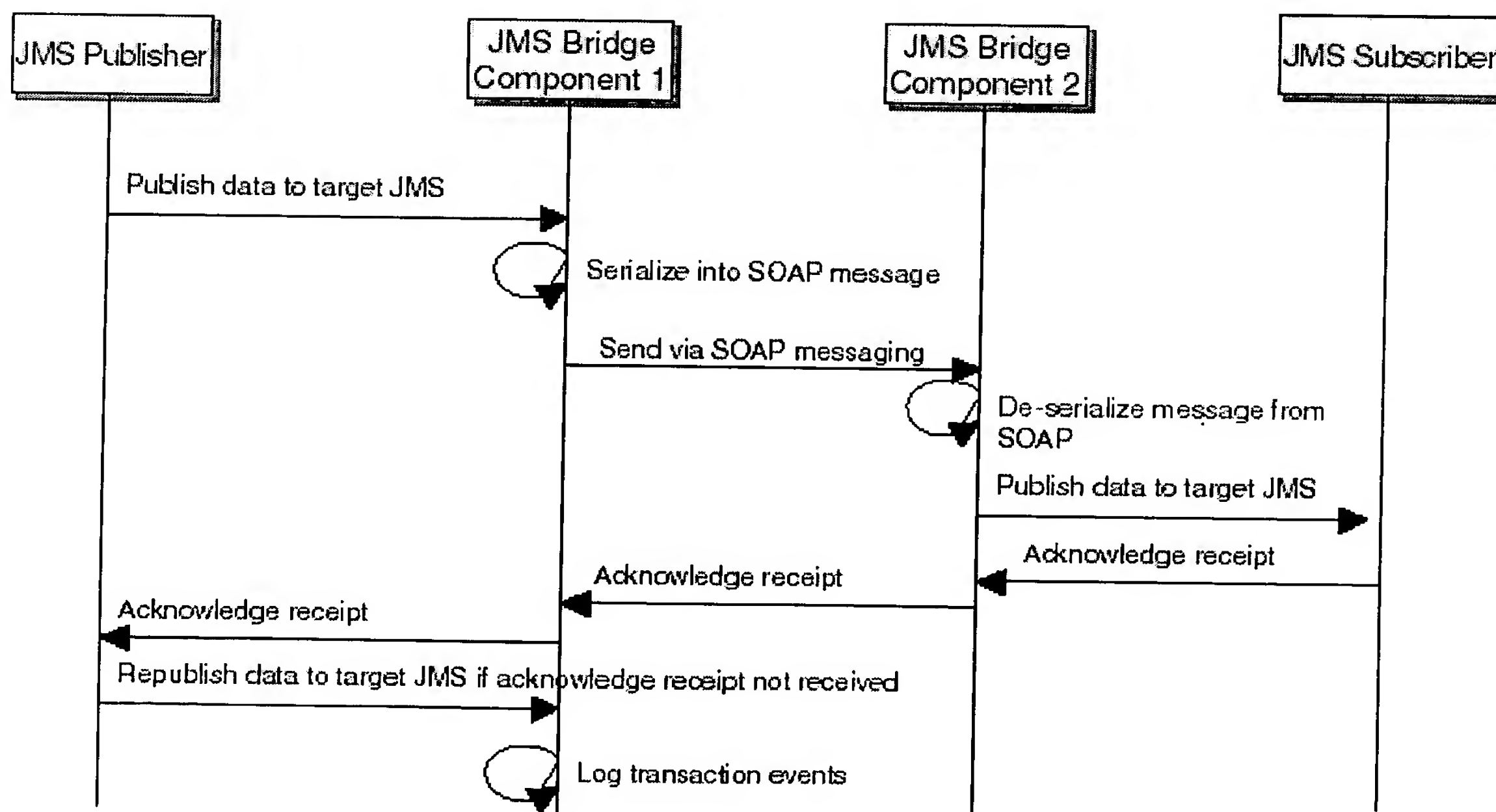


FIG. 29

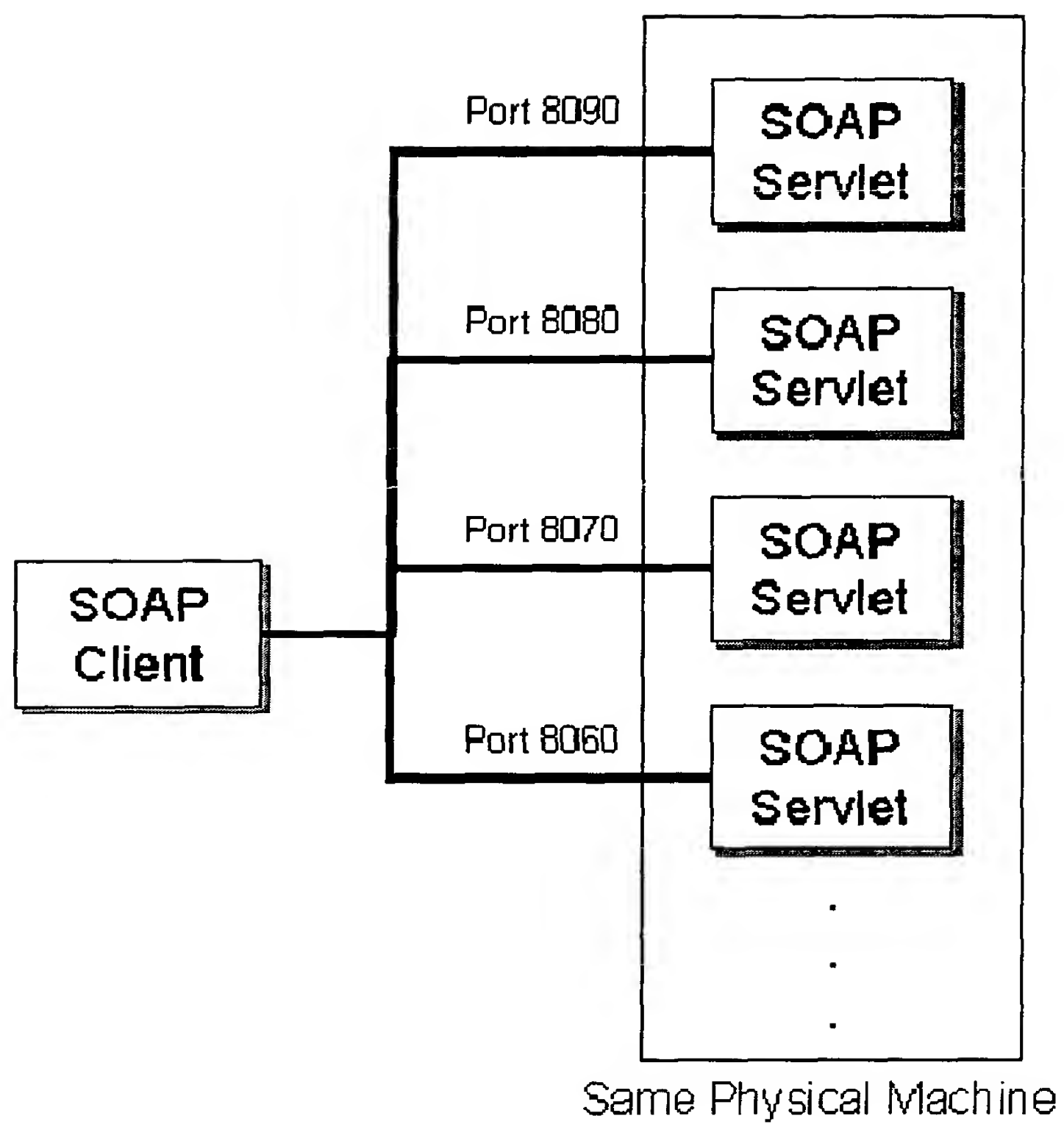


FIG. 30

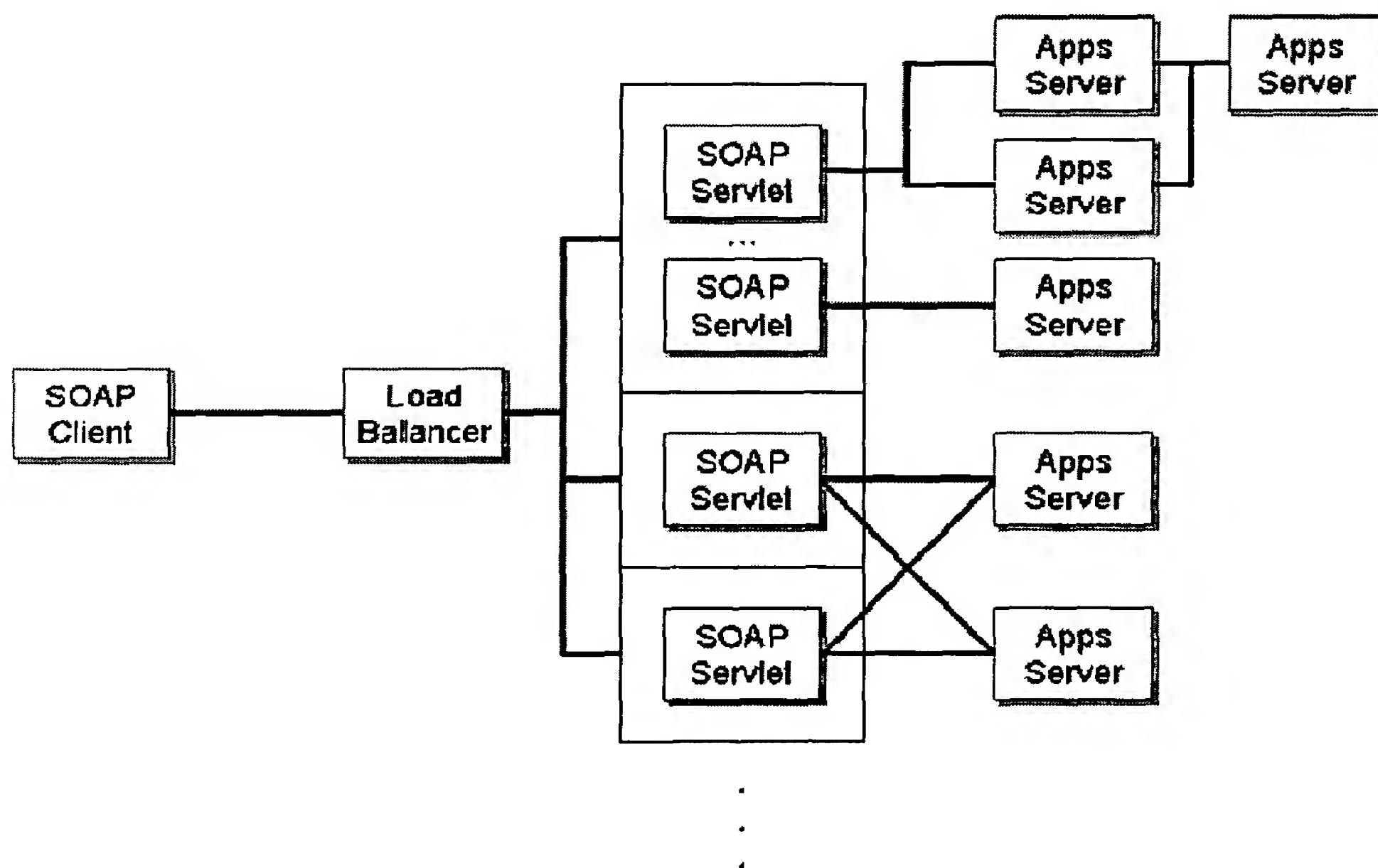
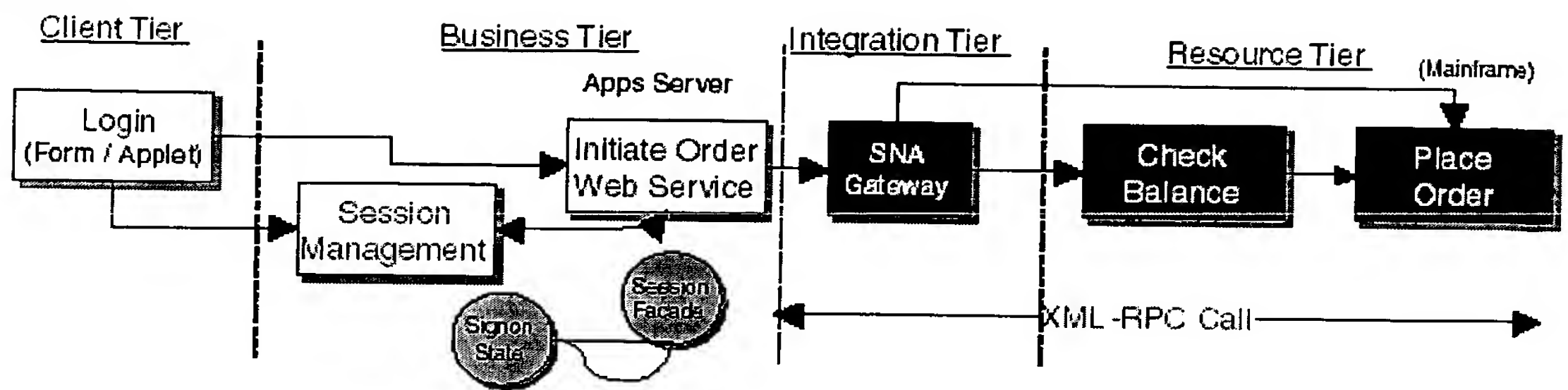


FIG. 31

# SCENARIO



# EXCEPTIONS

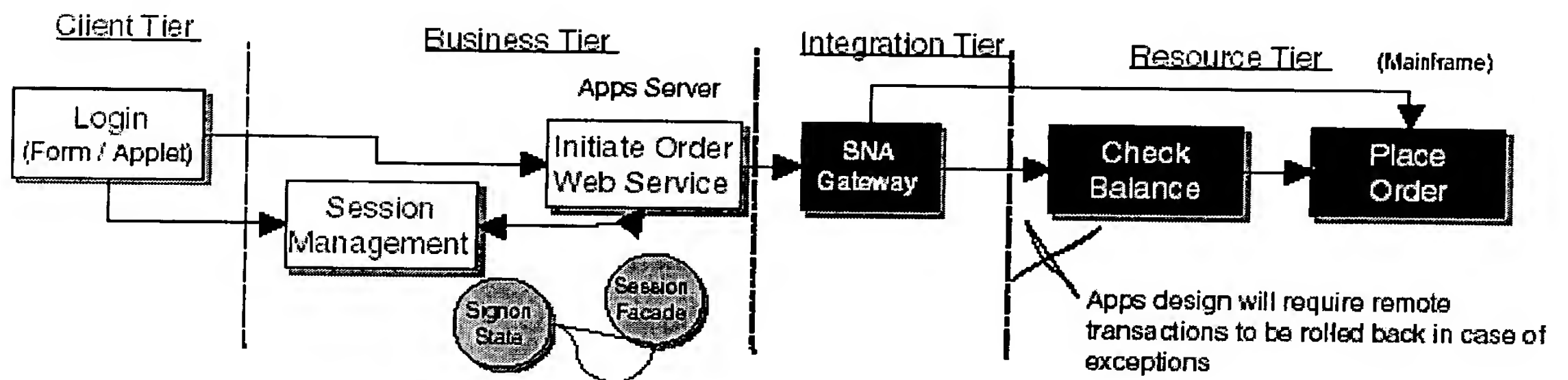


FIG. 32

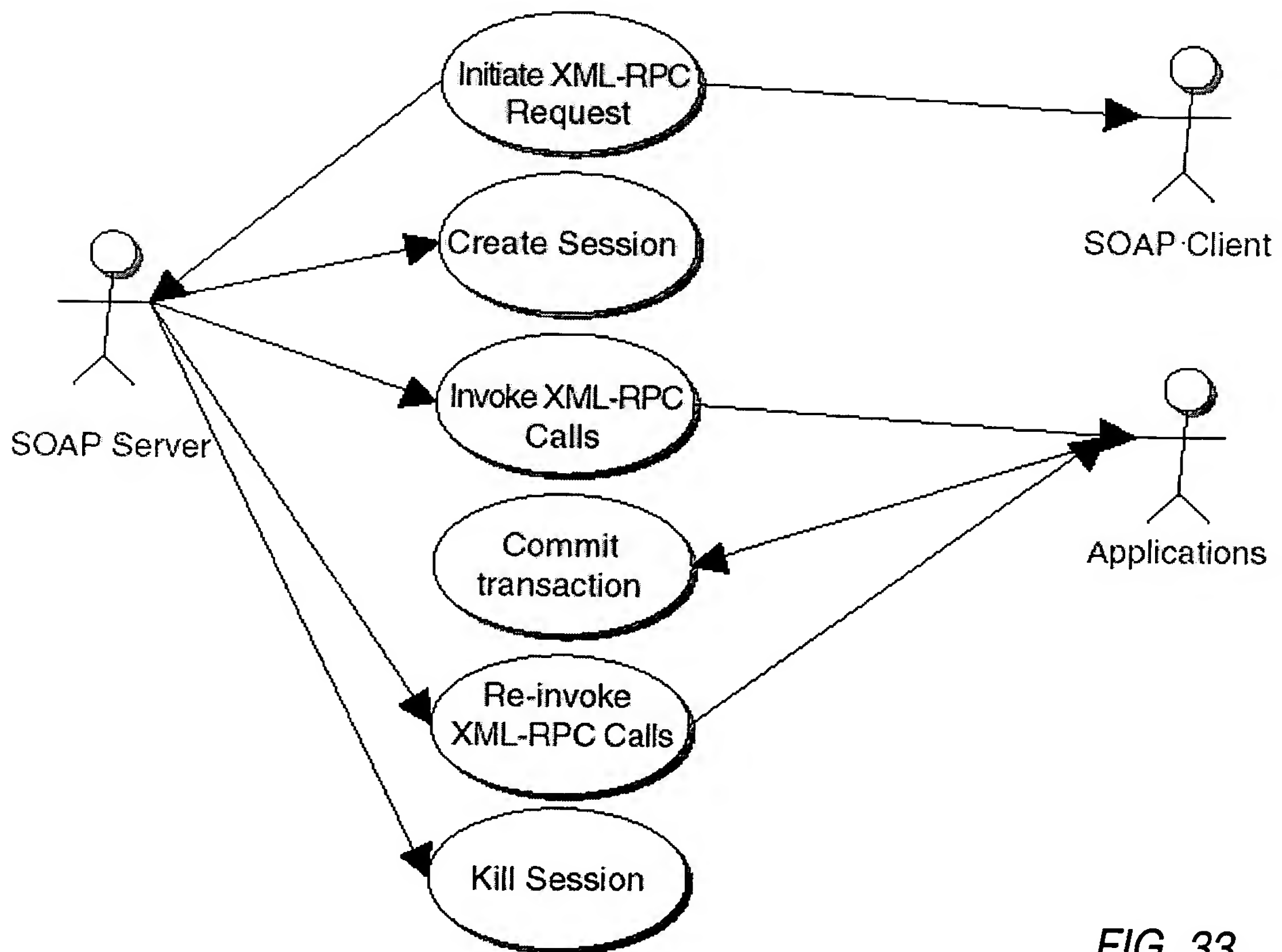


FIG. 33



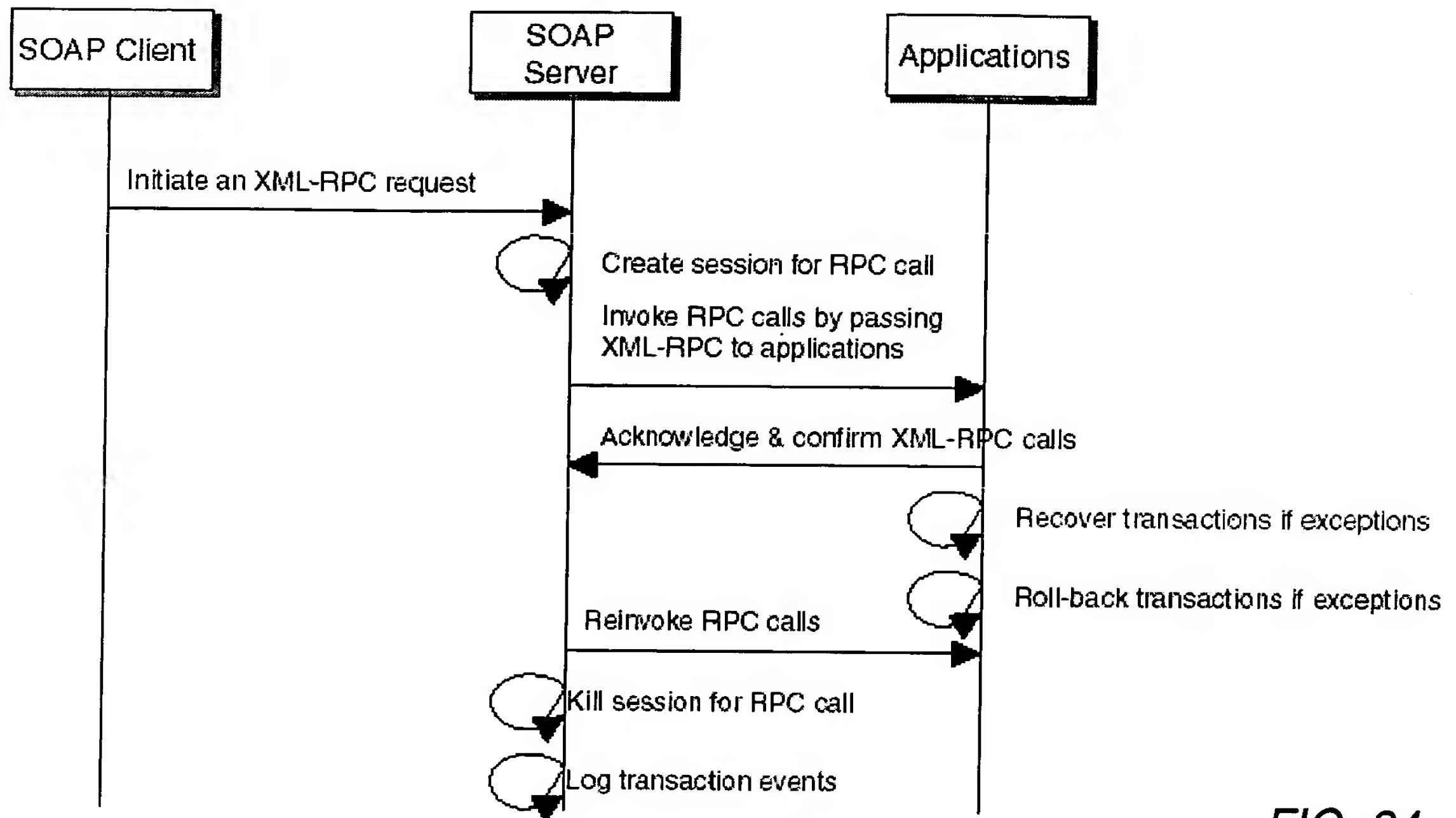


FIG. 34

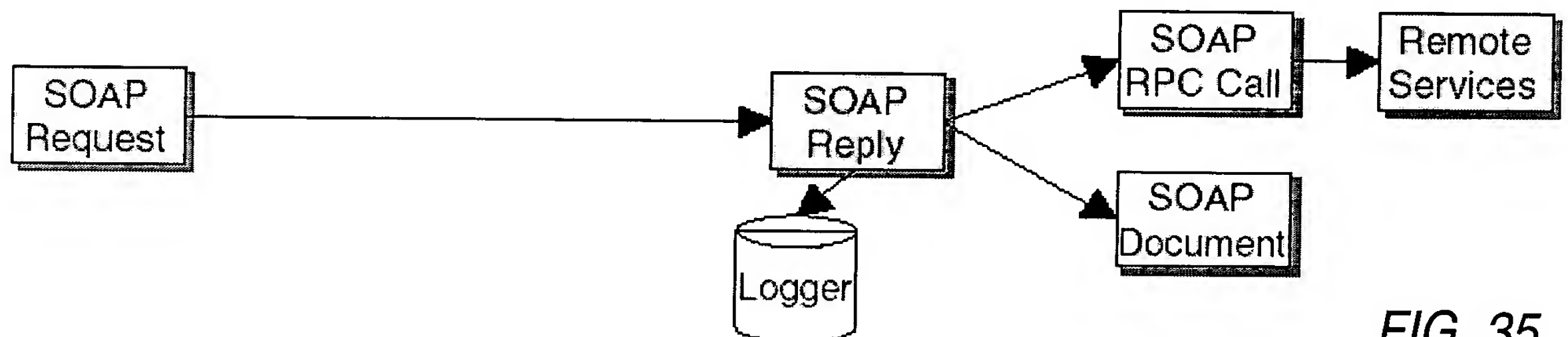


FIG. 35

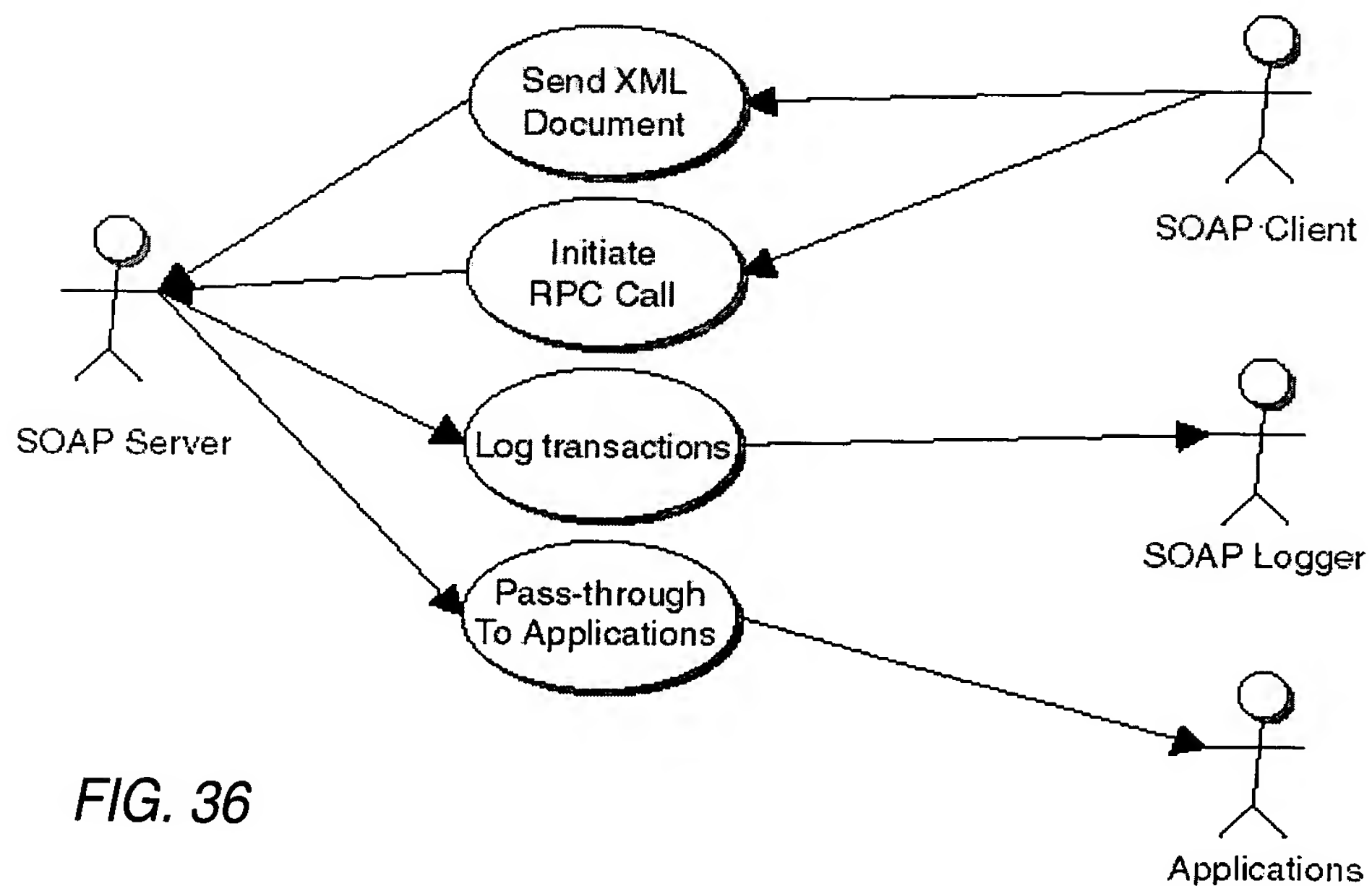


FIG. 36

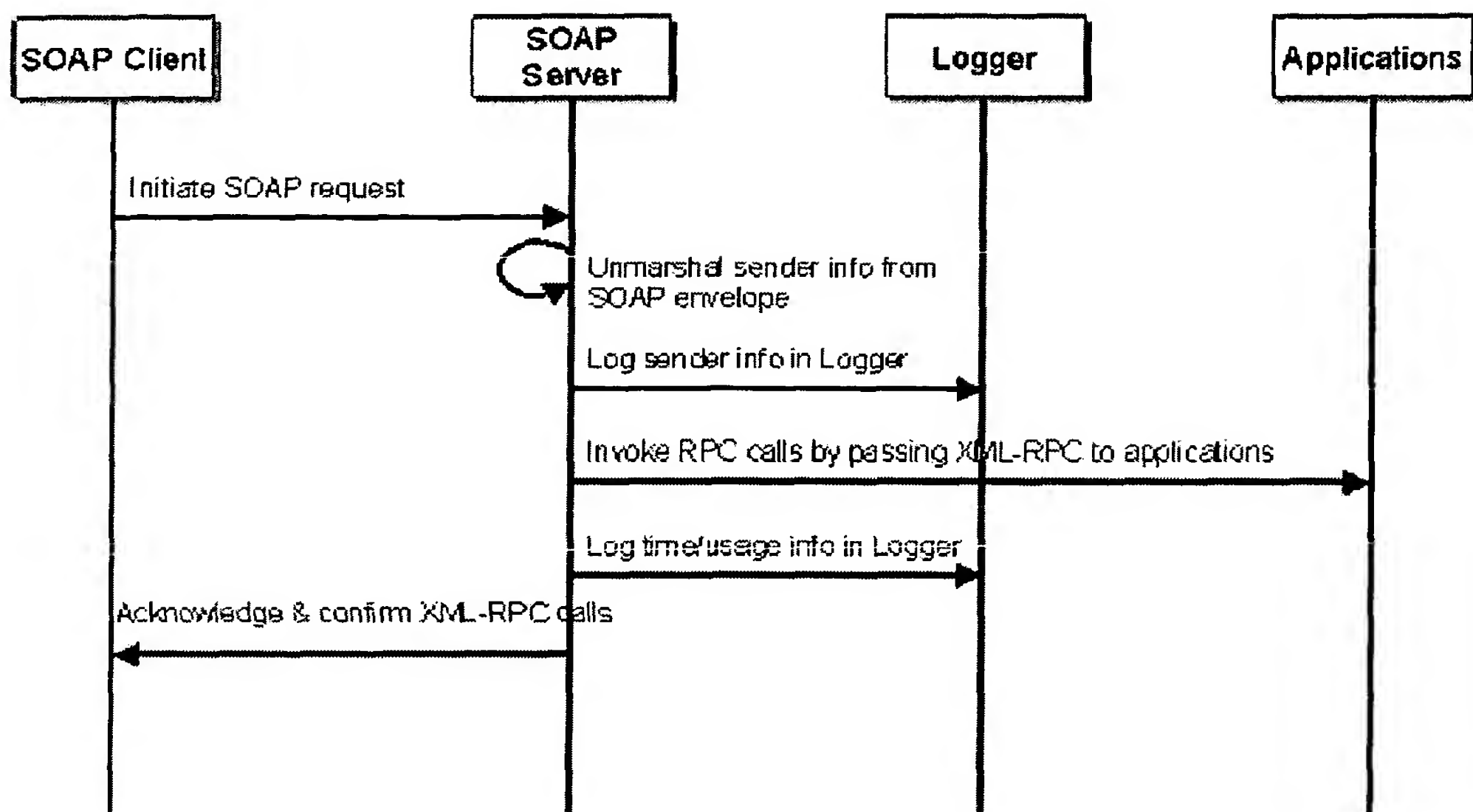


FIG. 37

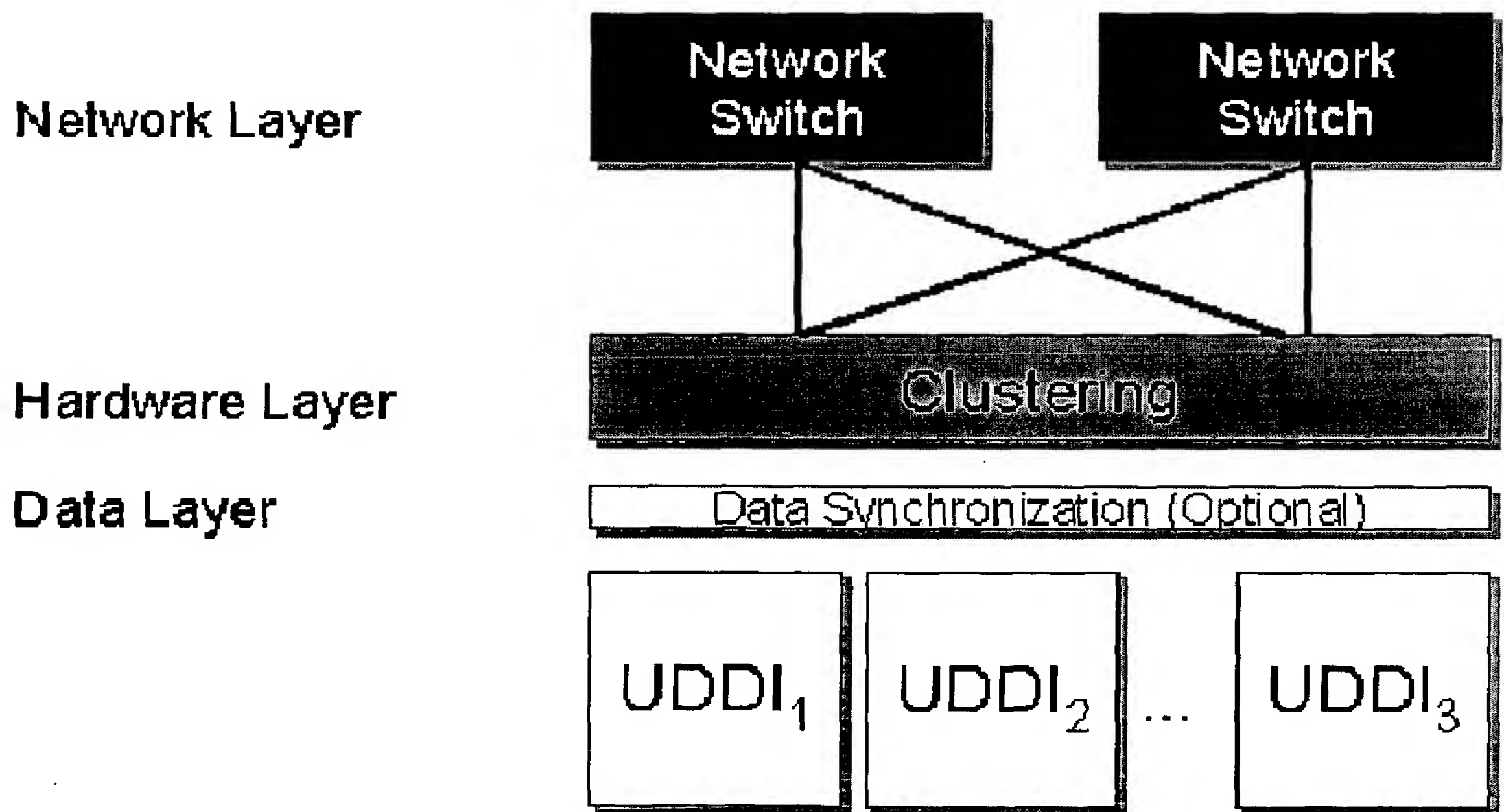
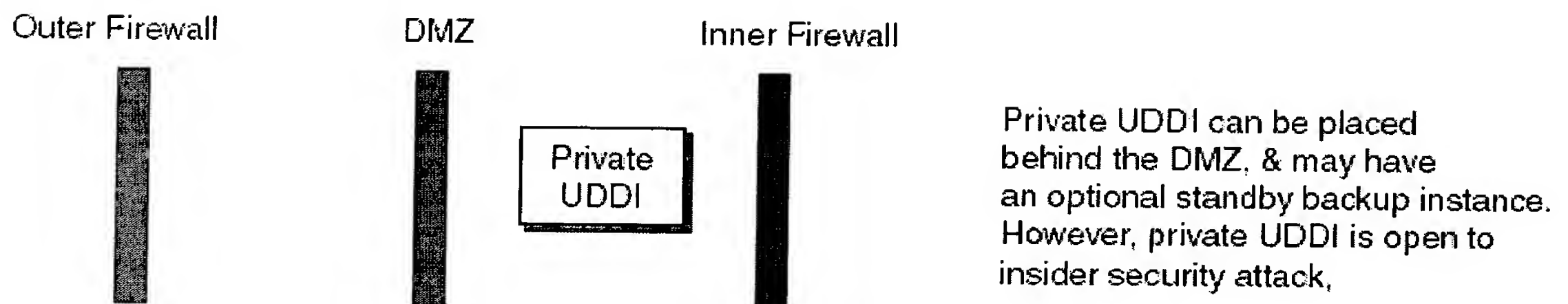
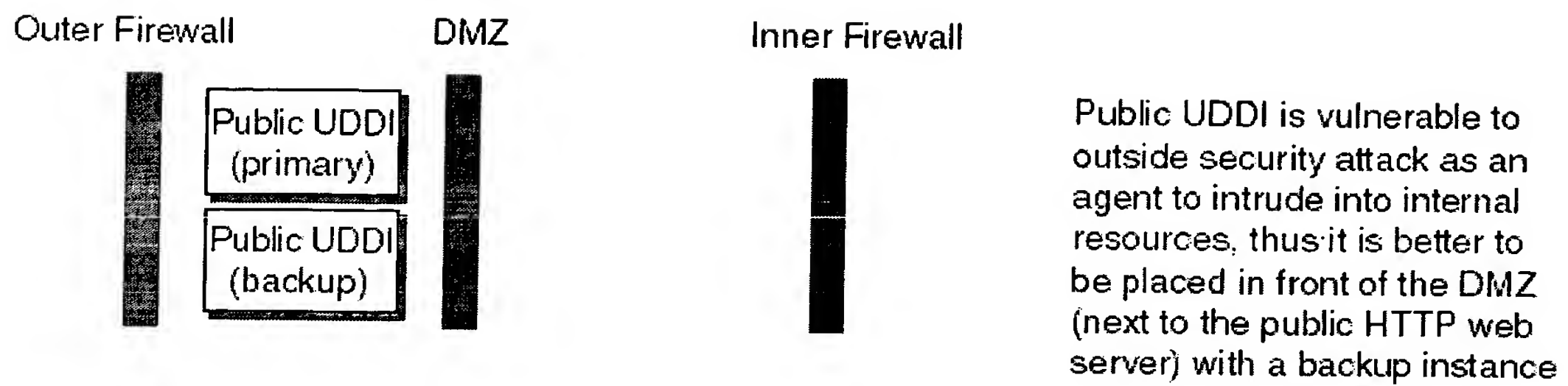
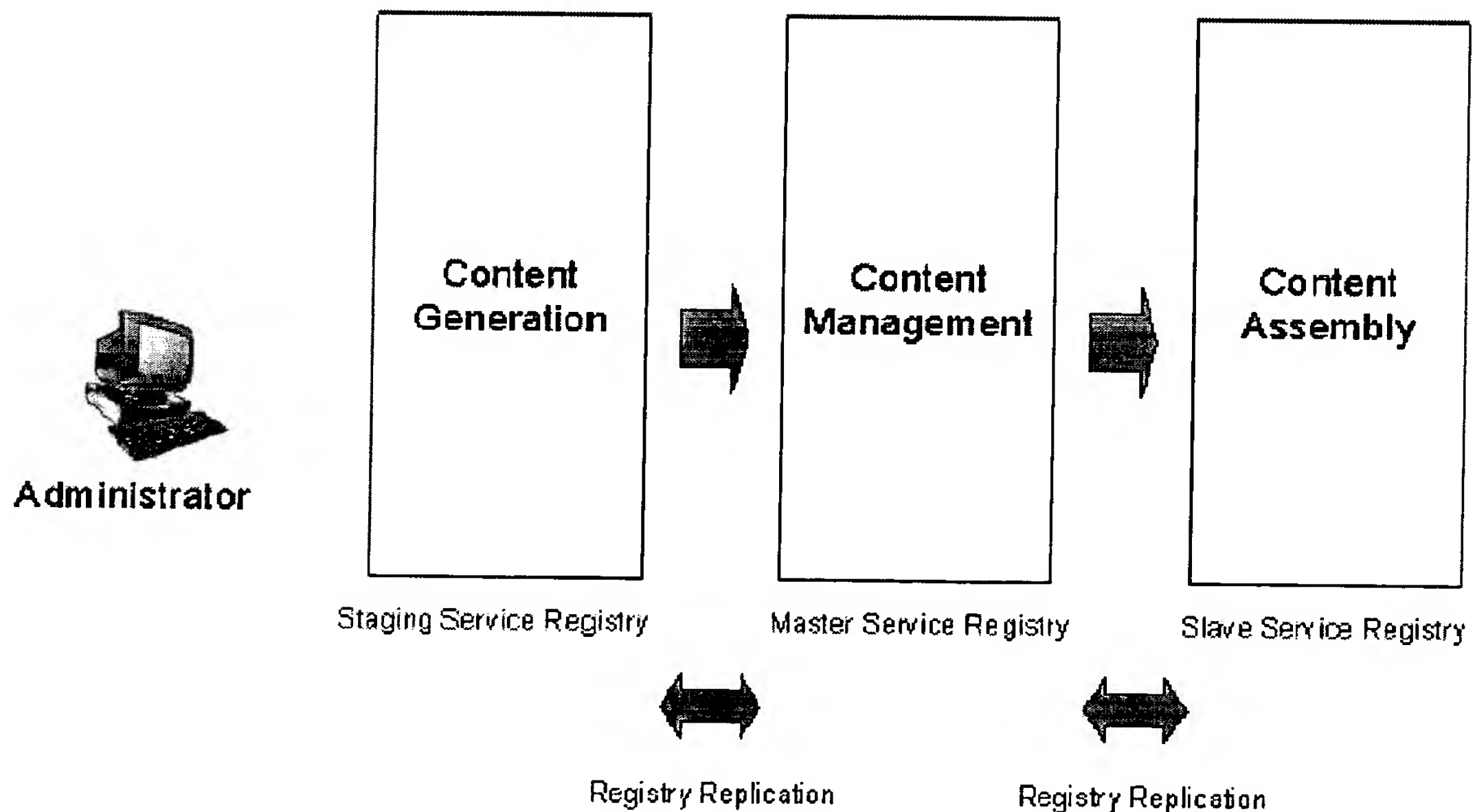


FIG. 38



**FIG. 39**



**FIG. 40**

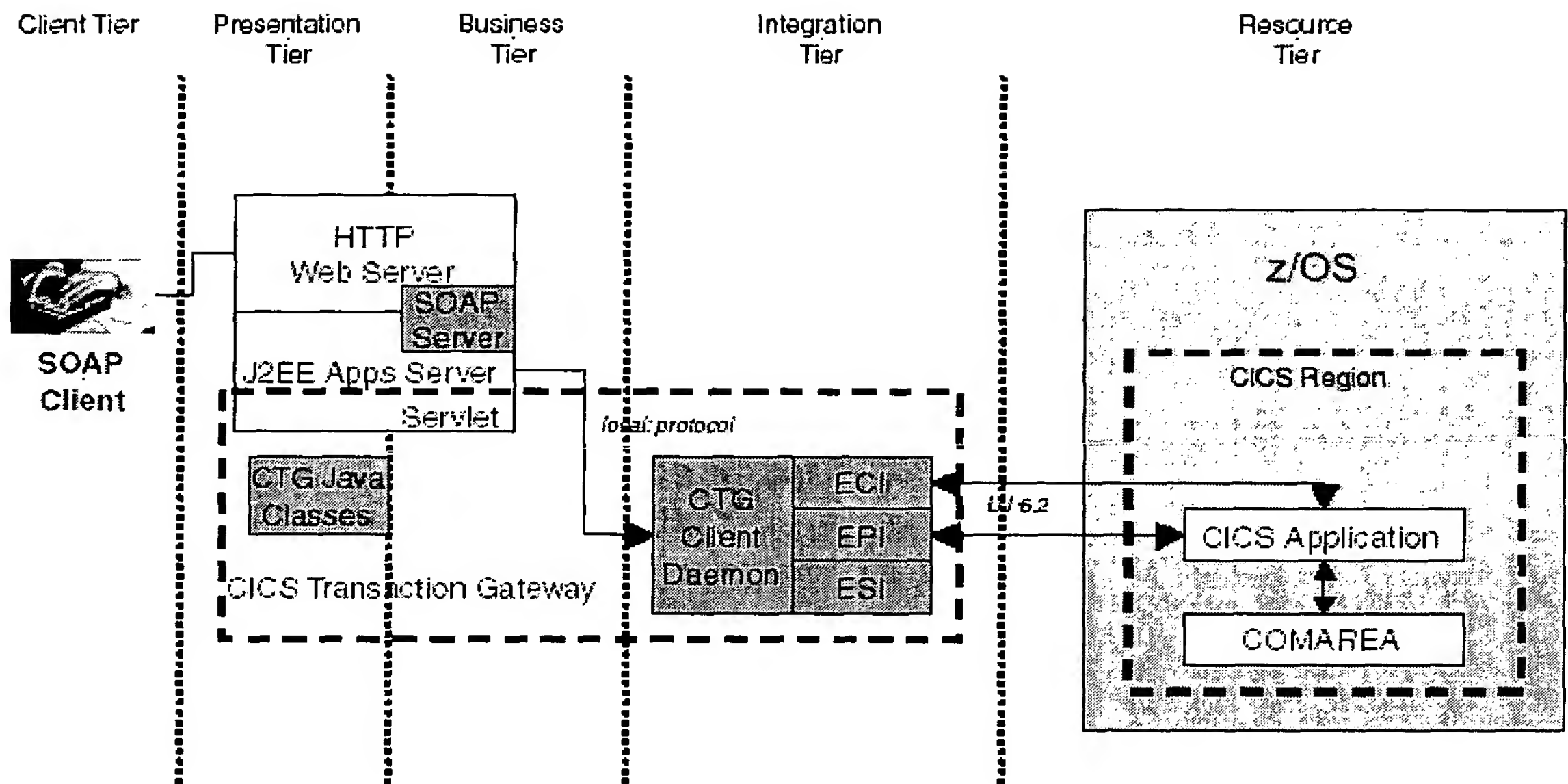


FIG. 41

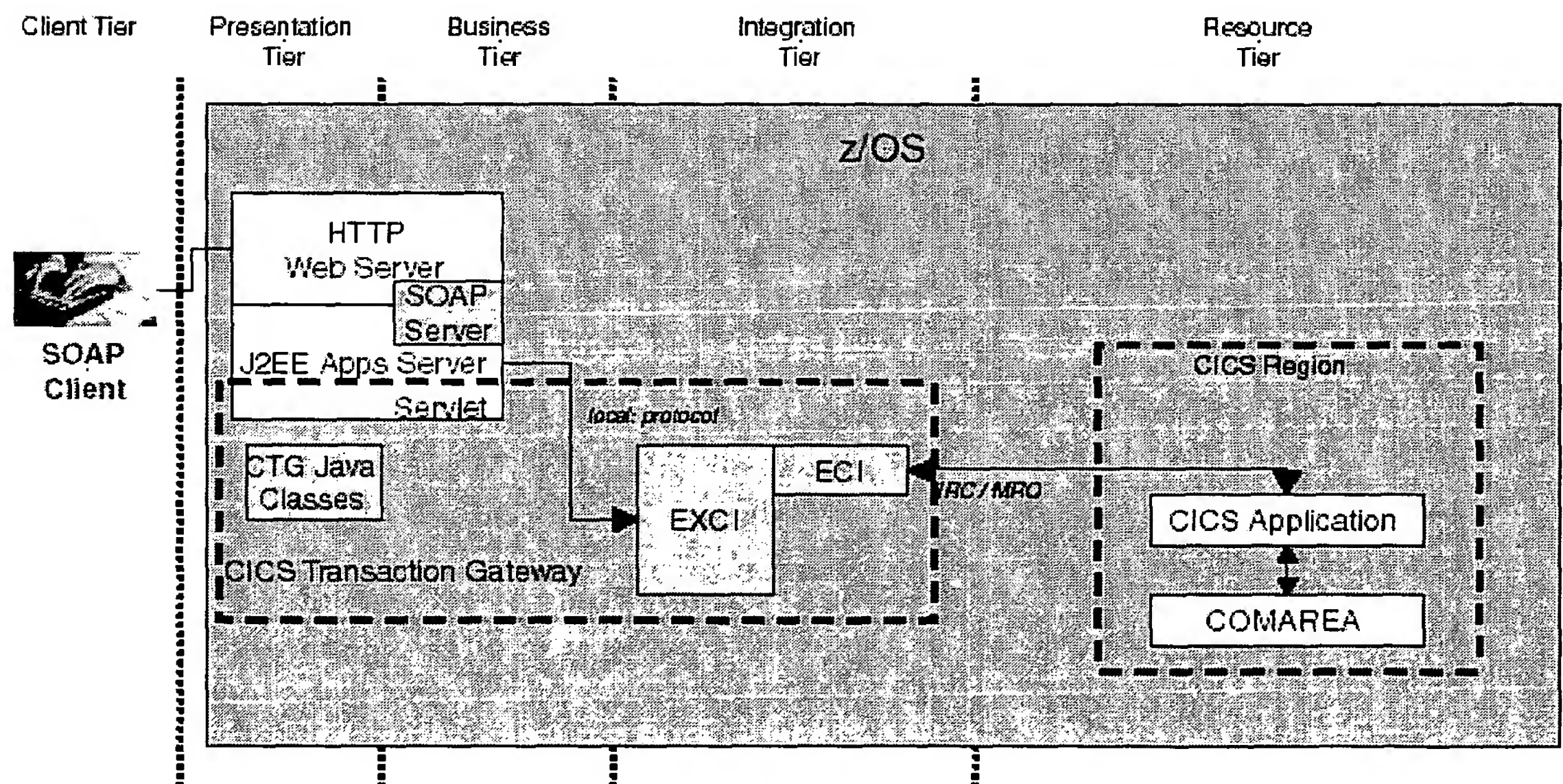


FIG. 42



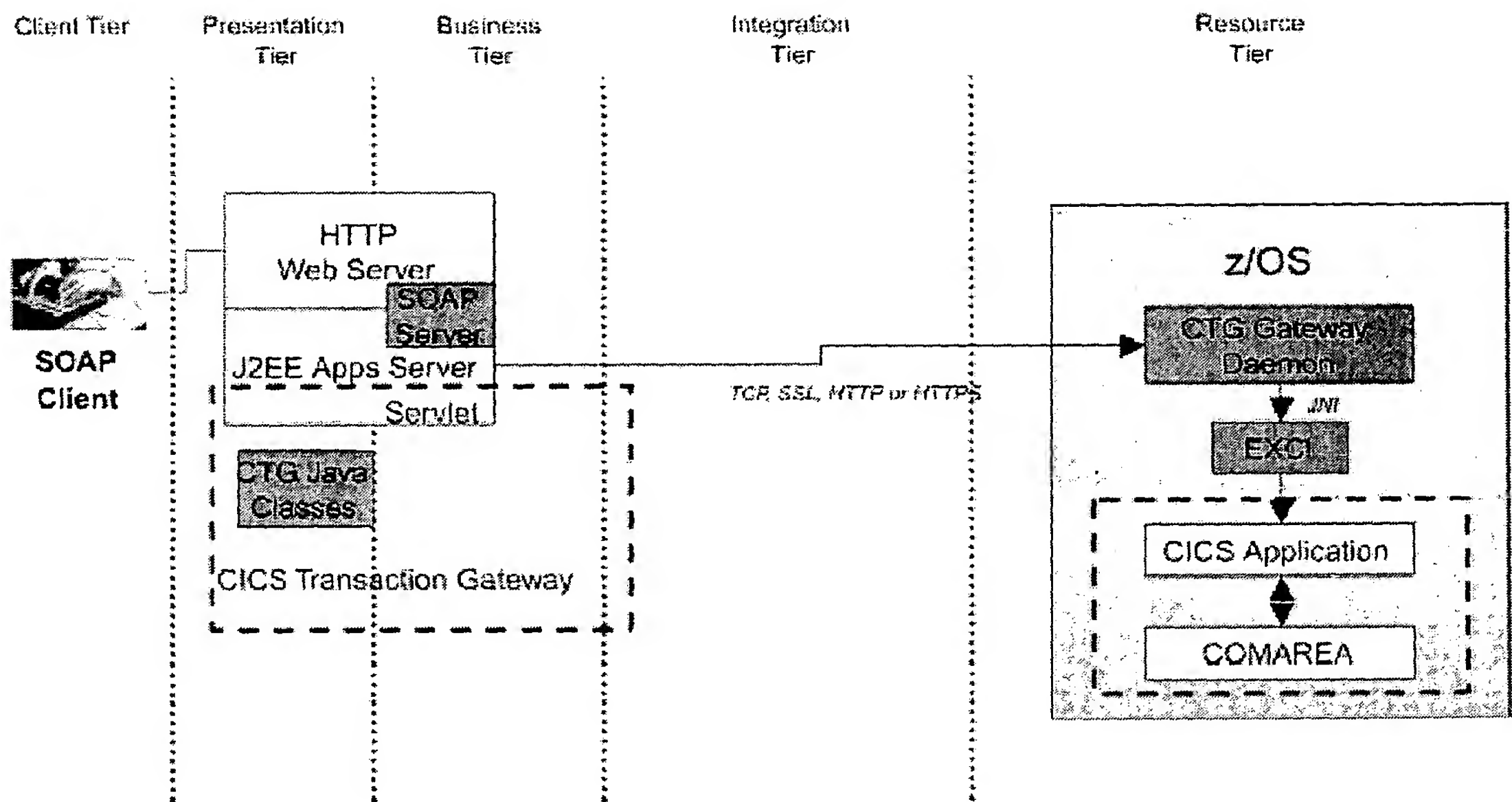


FIG. 43

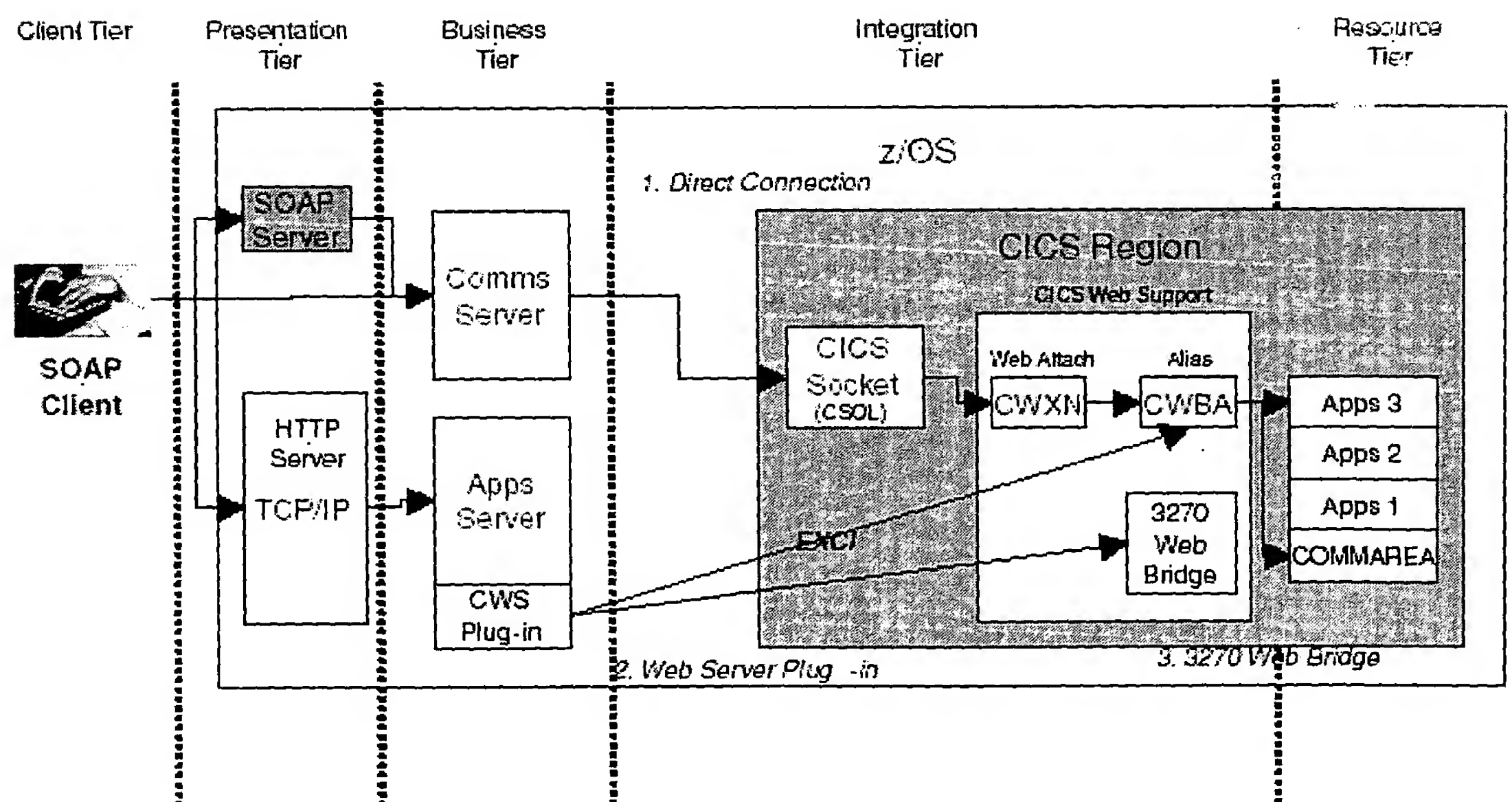


FIG. 44

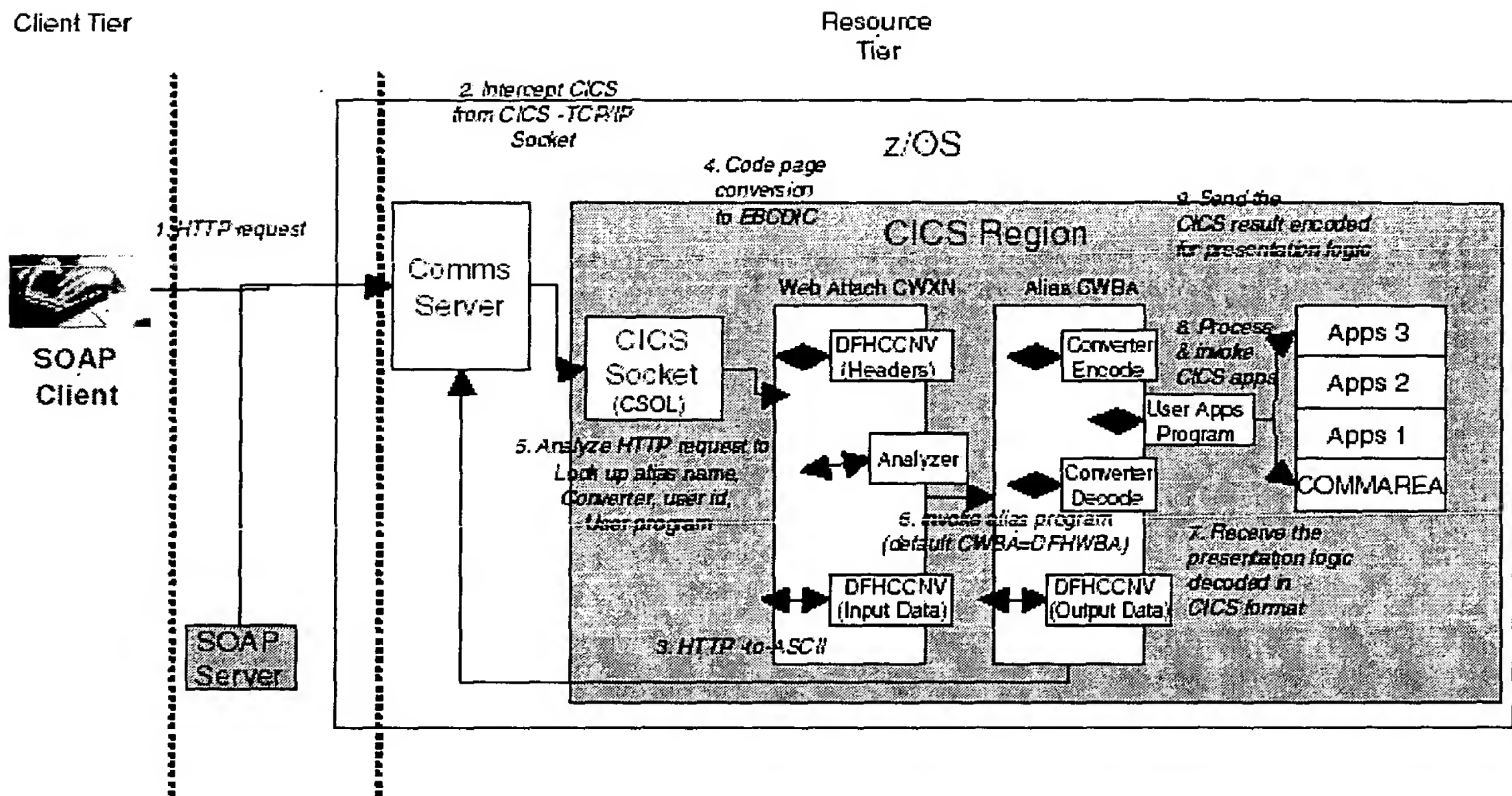


FIG. 45

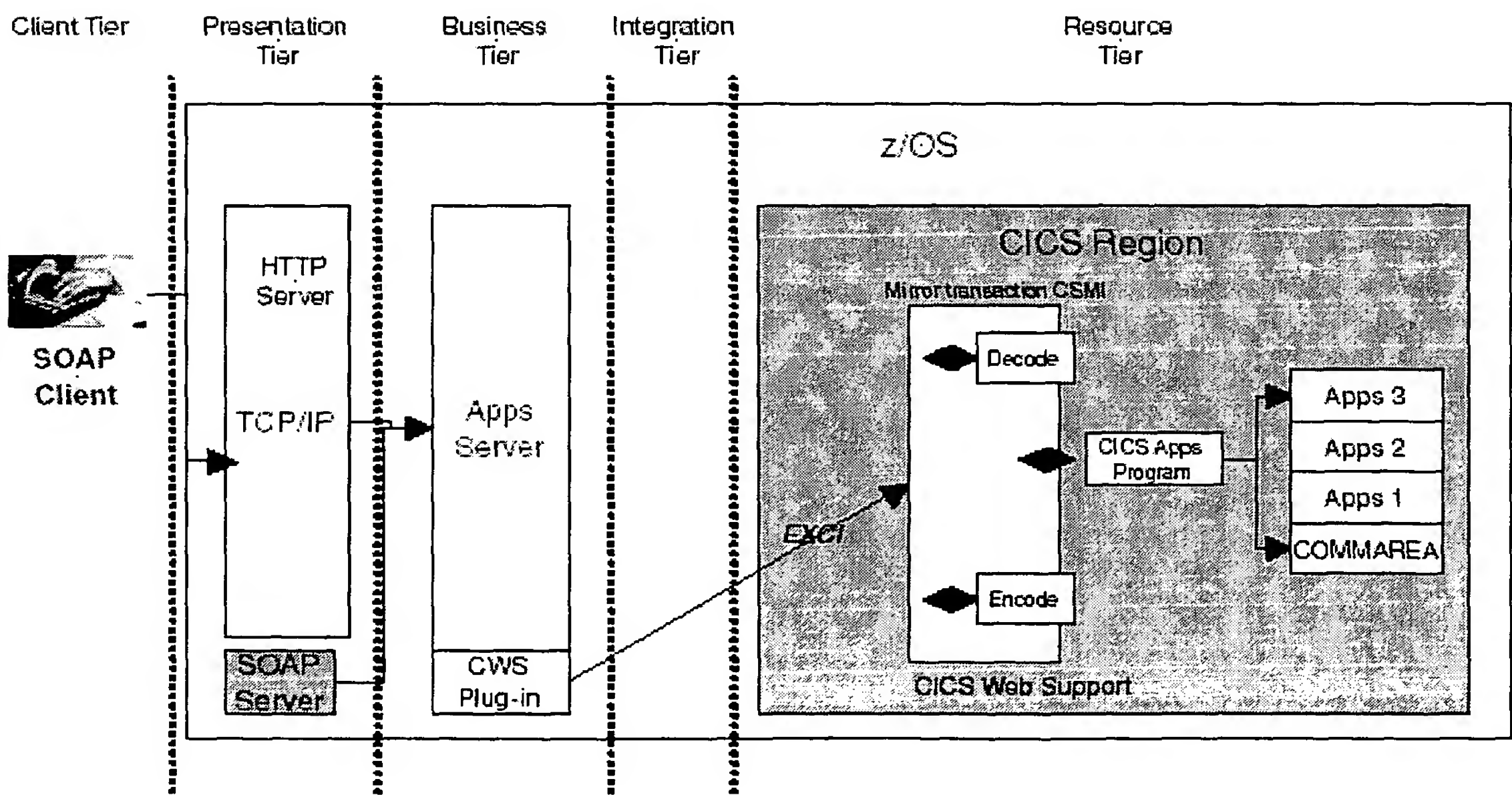


FIG. 46



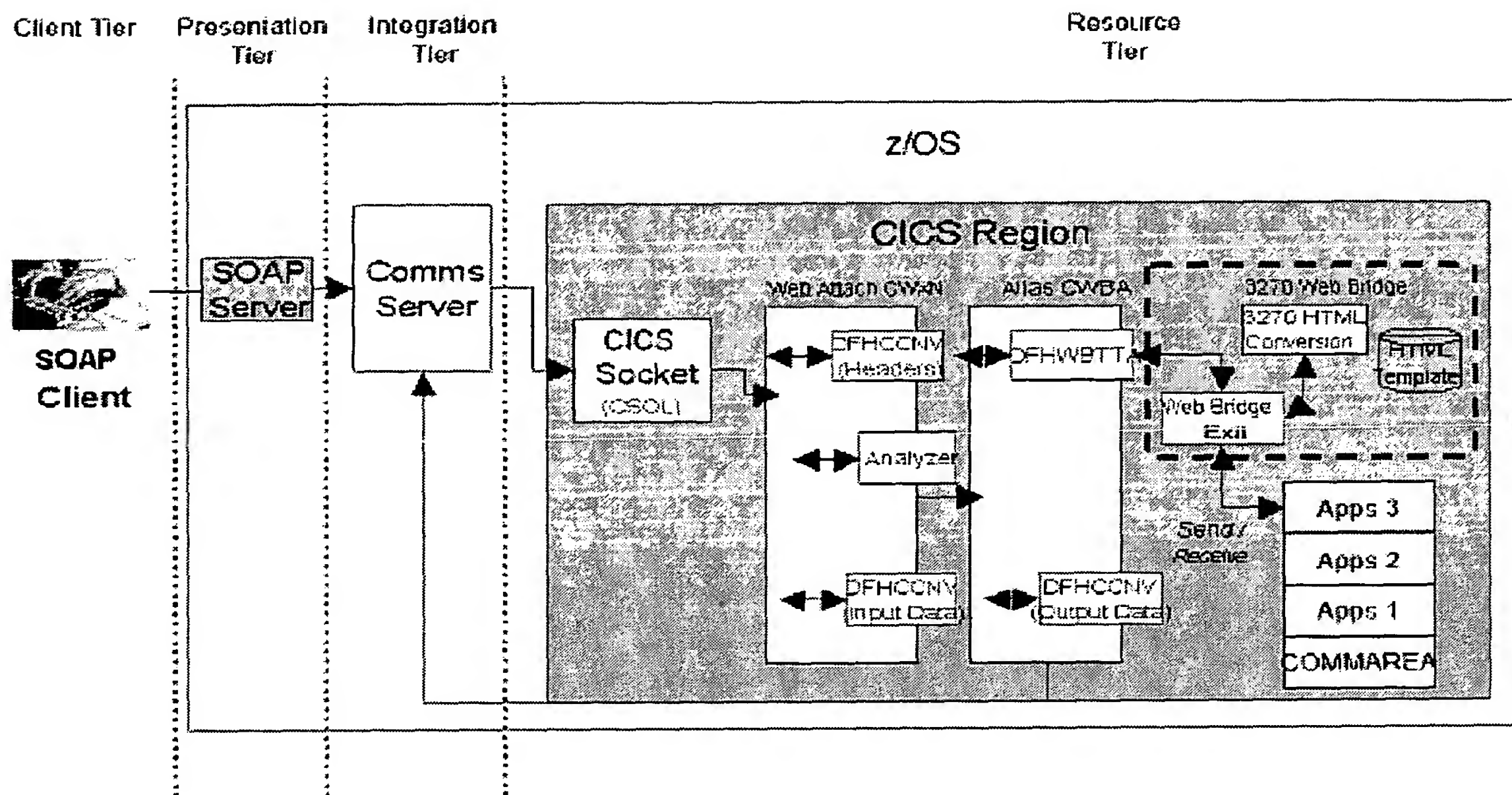


FIG. 47

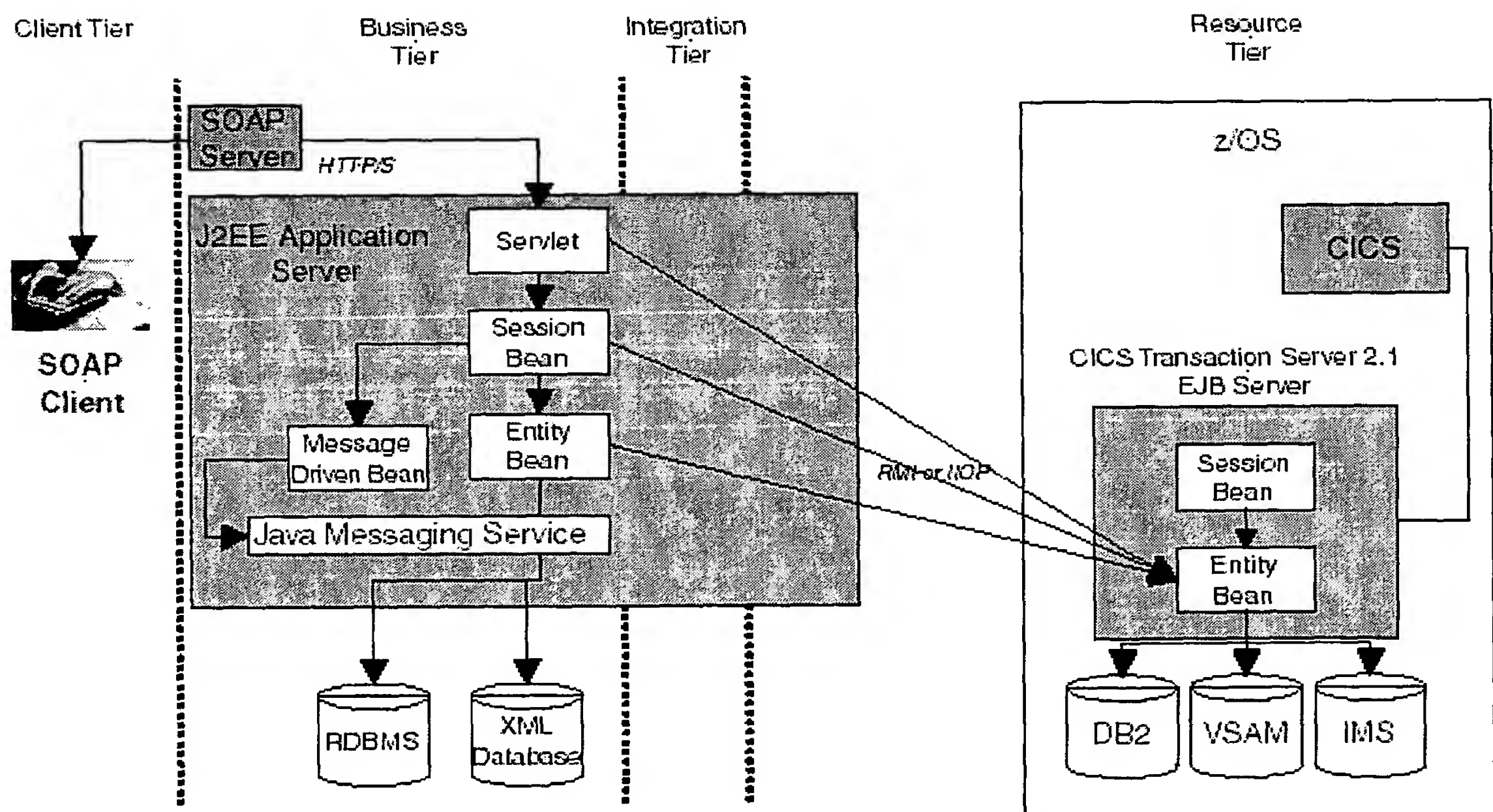


FIG. 48



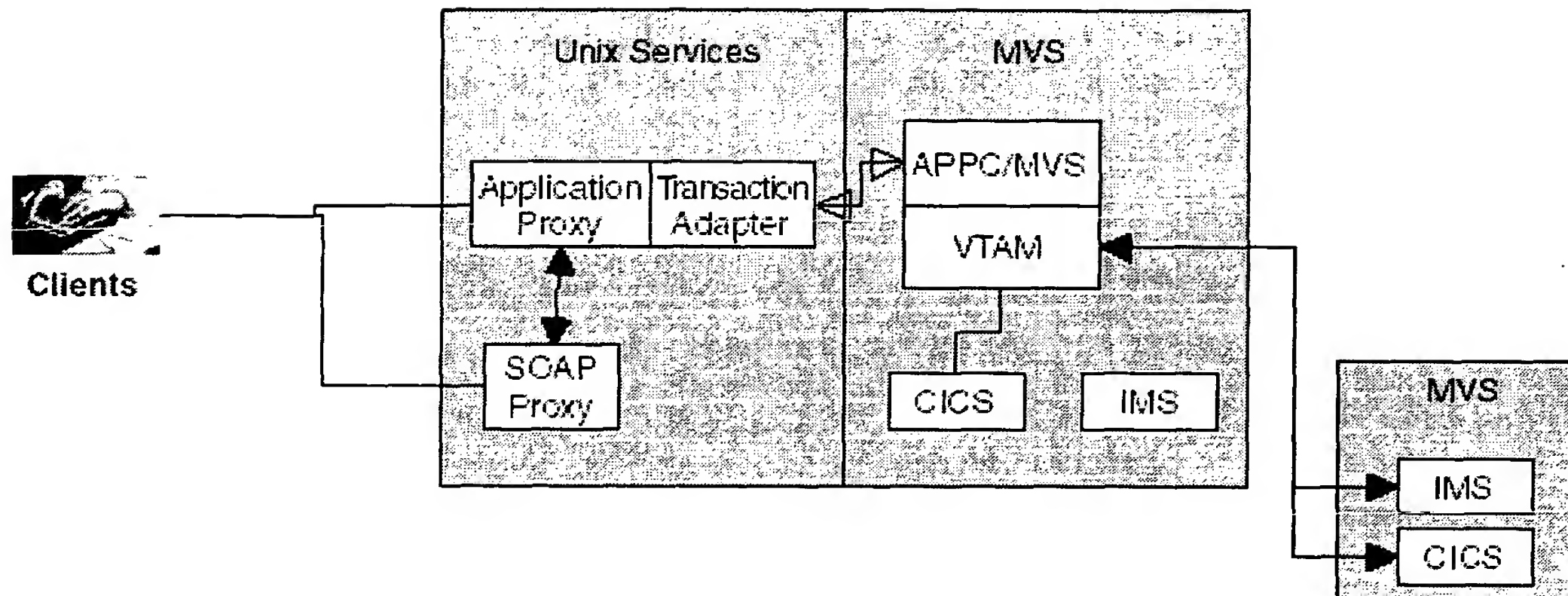


FIG. 49

Technology Approach	Business Tier (Application Server)	Integration Tier	Resource Tier (Back-End Legacy System)
CICS Transaction Gateway		CICS Transaction Gateway—use of ECI, EPI, and ESI calls	
CICS Web Support			CICS Web Support—using CWS to Web-enable 3270-based CICS applications
Java	Enterprise Java Beans—abstracting business functionality from legacy systems	Java Connector Architecture—standardizing connectors to legacy systems	CICS EJB Server—EJB container to support EJB
SOAP Proxy on Mainframe		Forte Transaction Adapter—building Application Proxy for back-end resources	Forte Transaction Adapter—server side for APPC conversation

FIG. 50

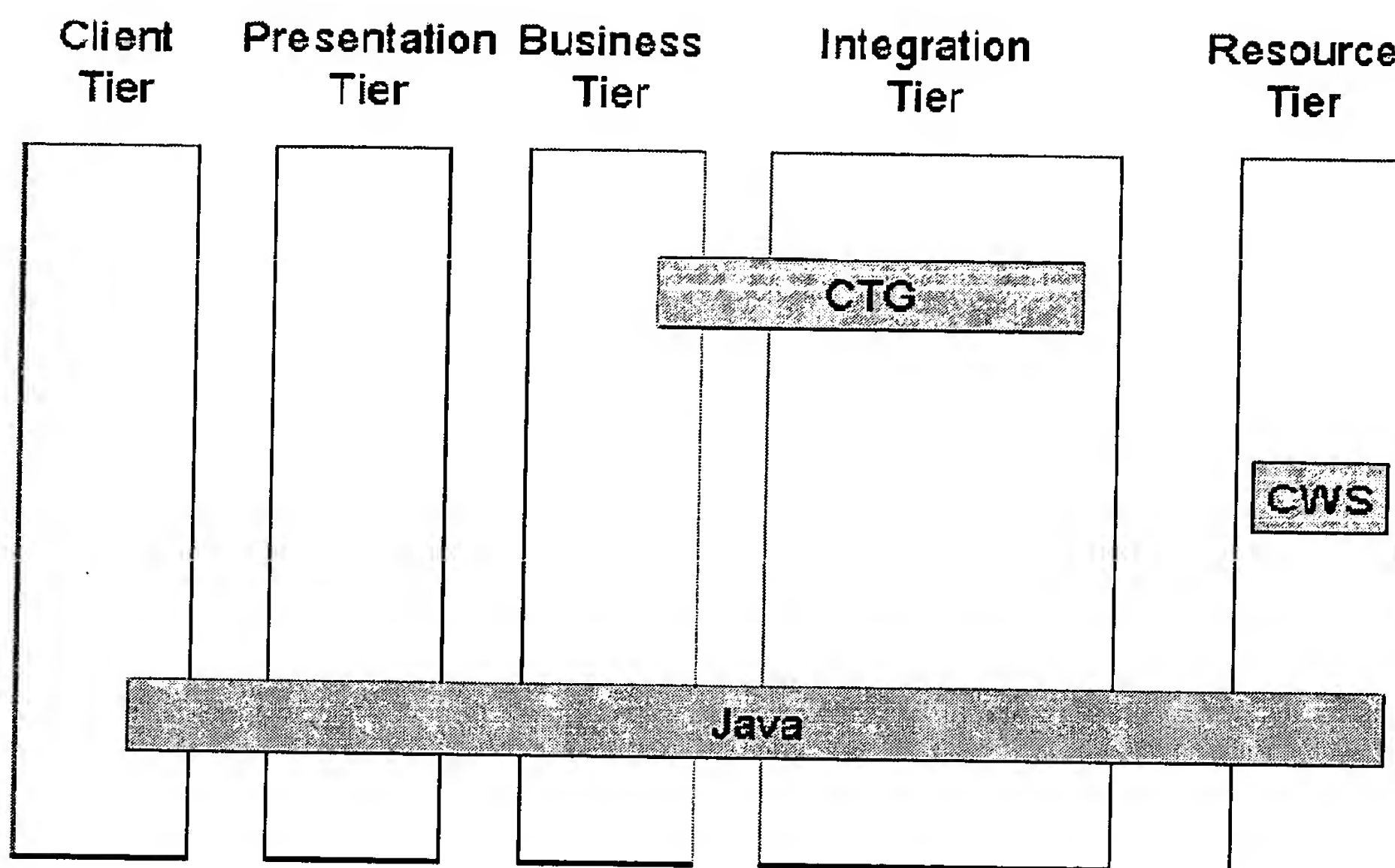


FIG. 51

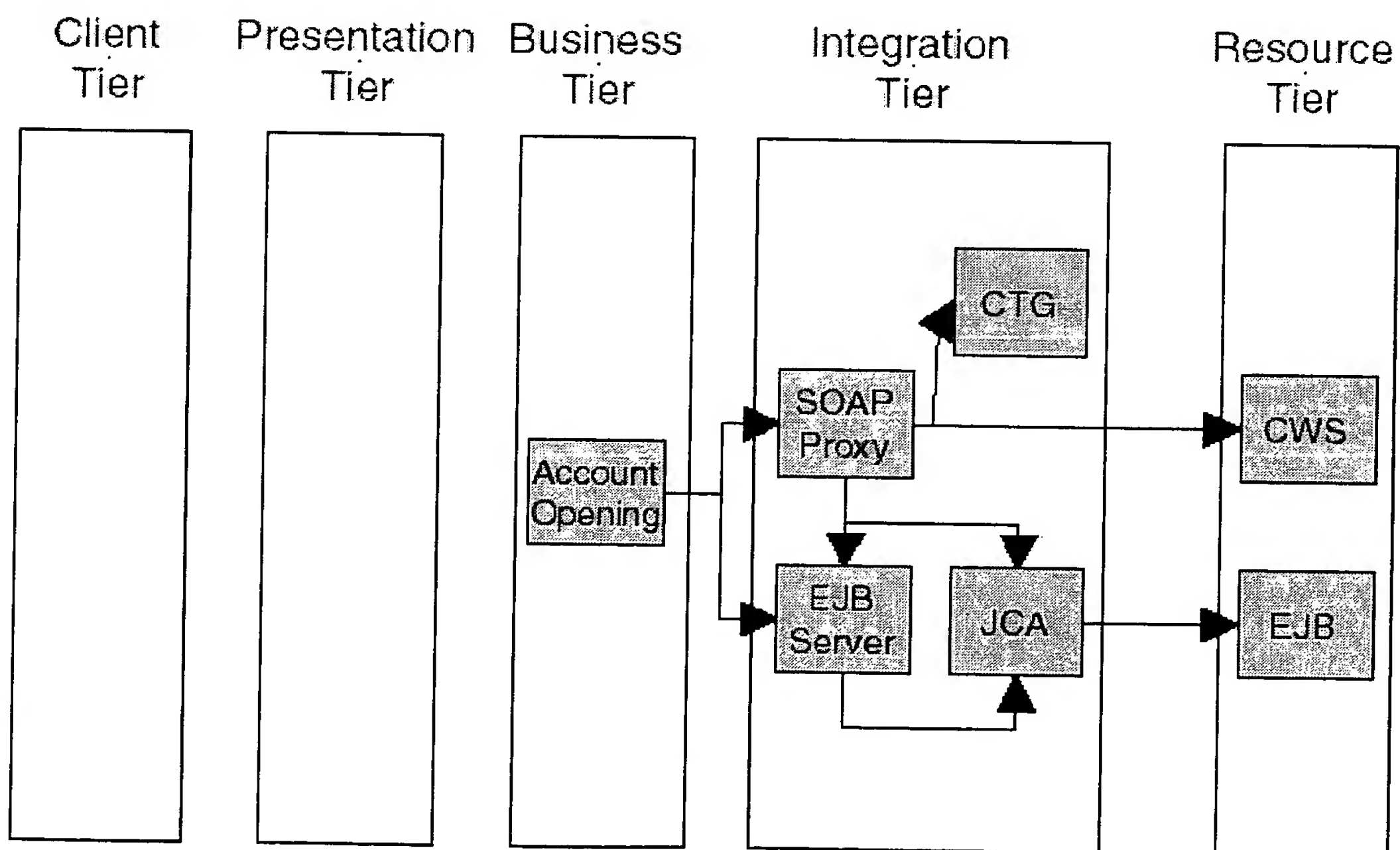


FIG. 52

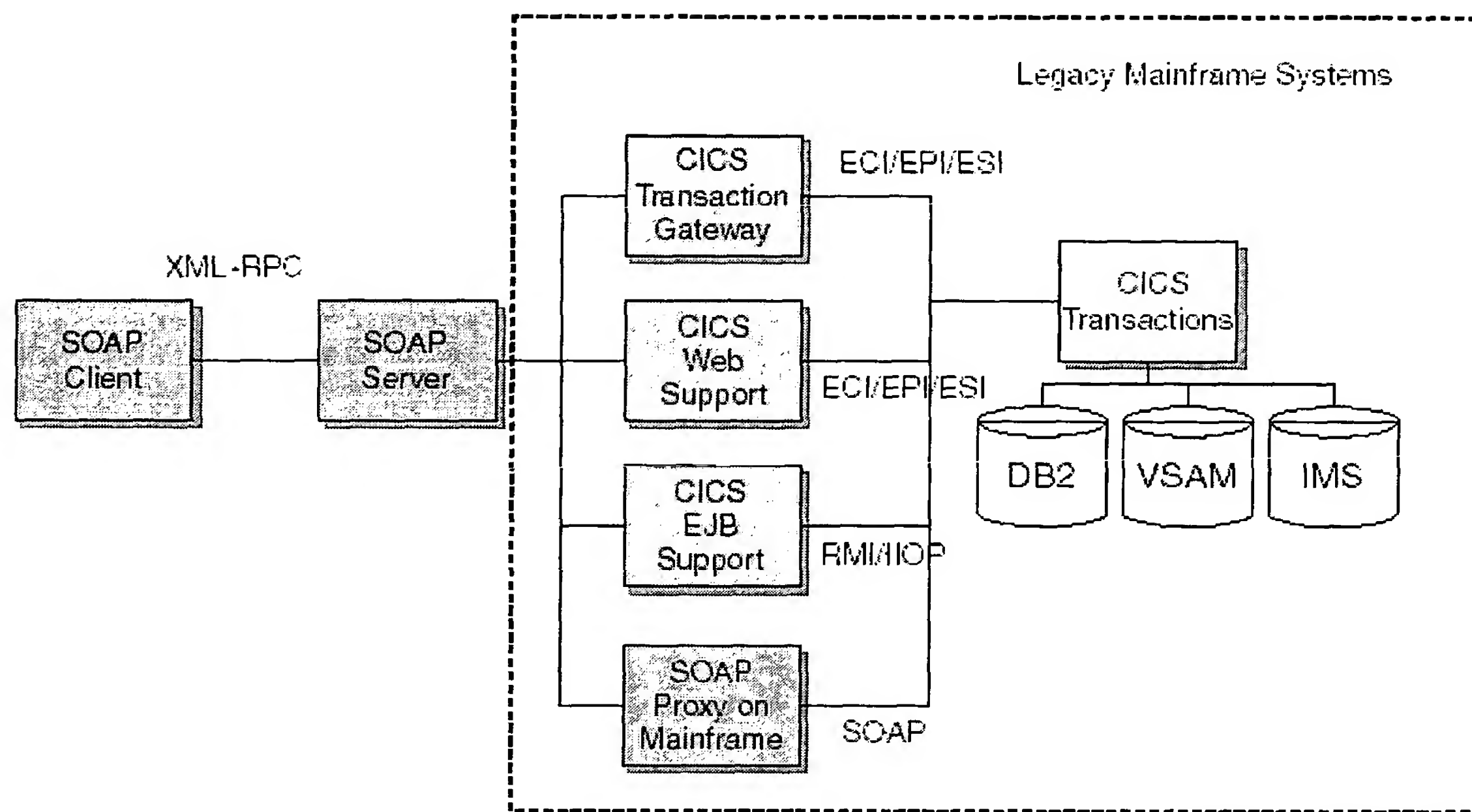


FIG. 53

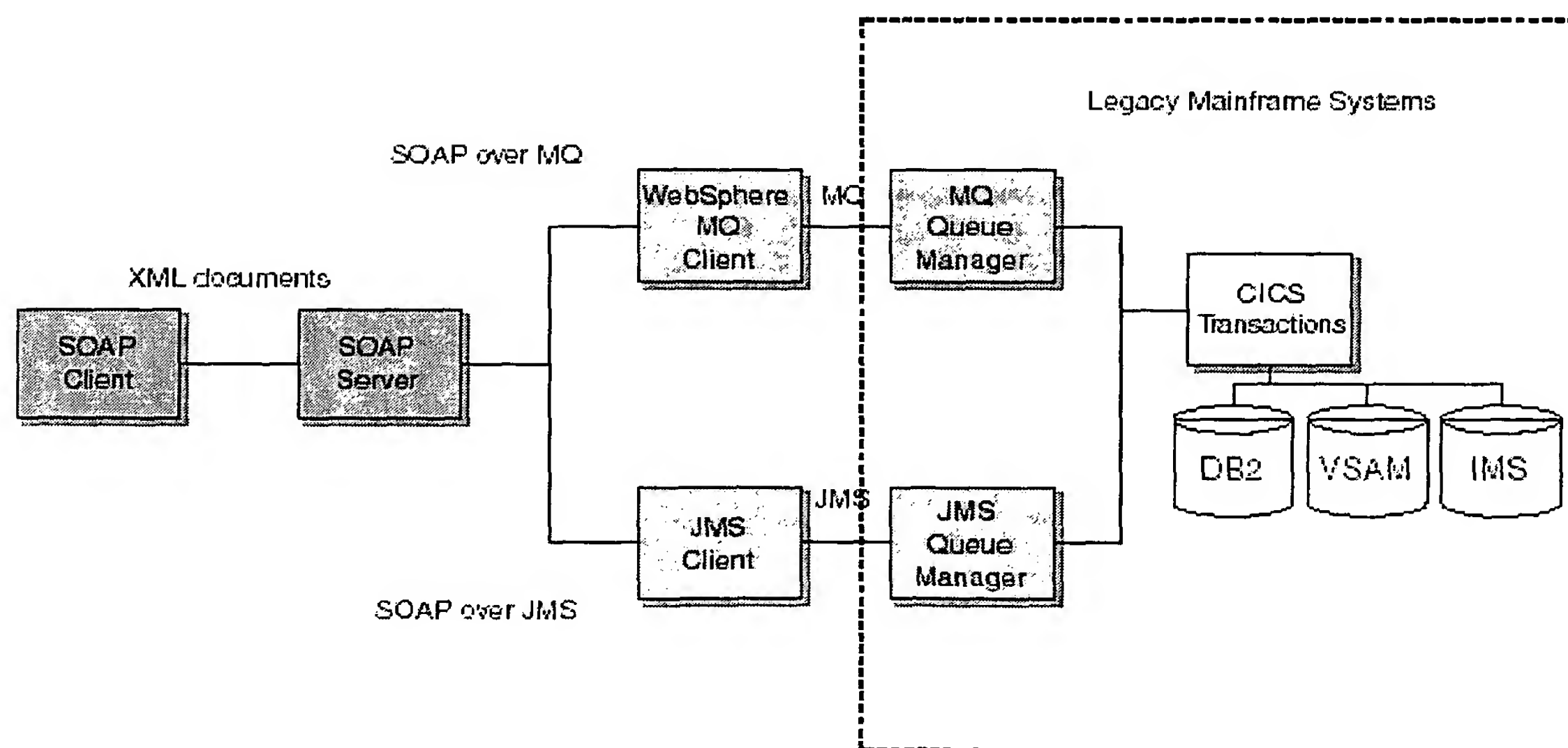


FIG. 54

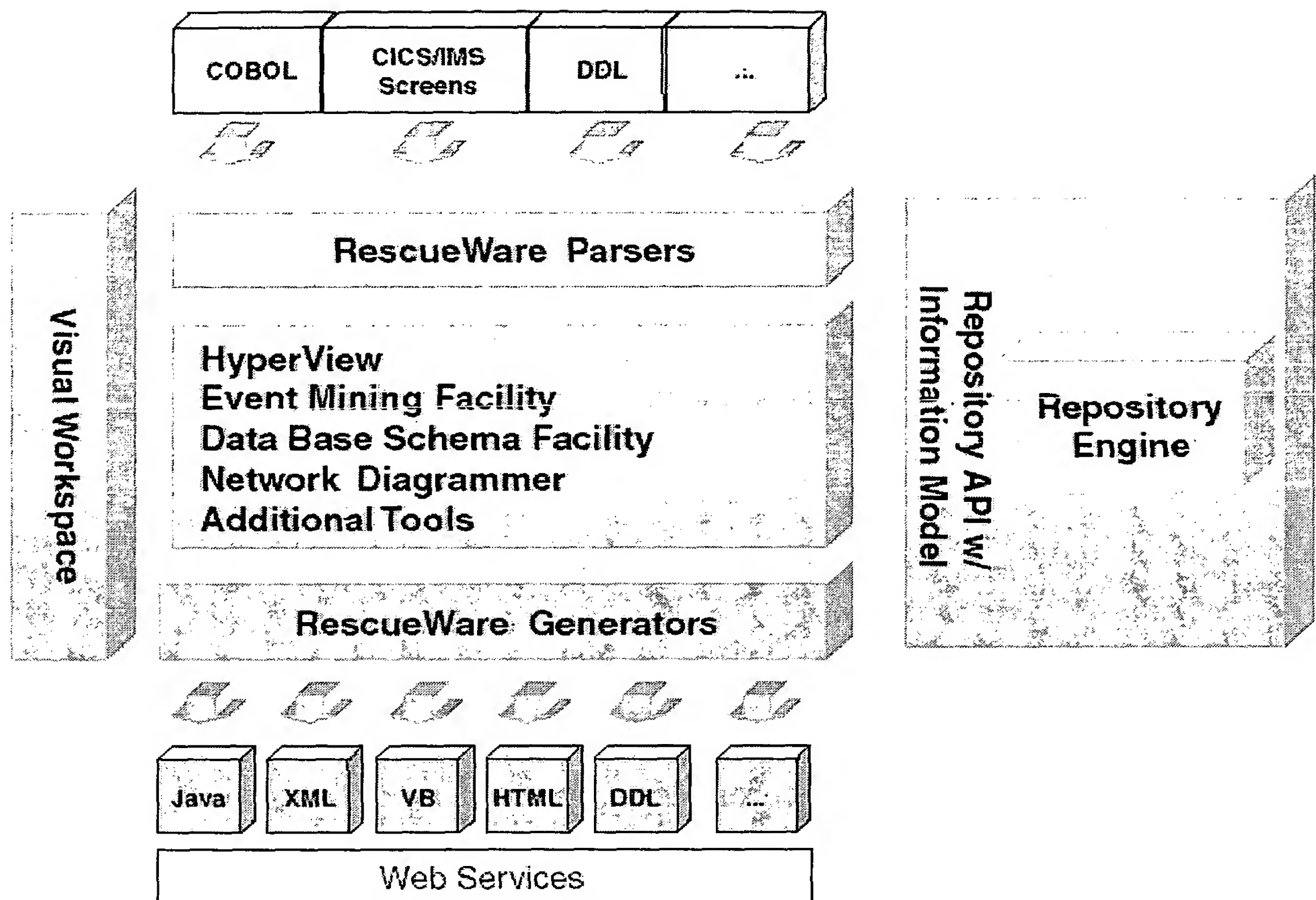


FIG. 55

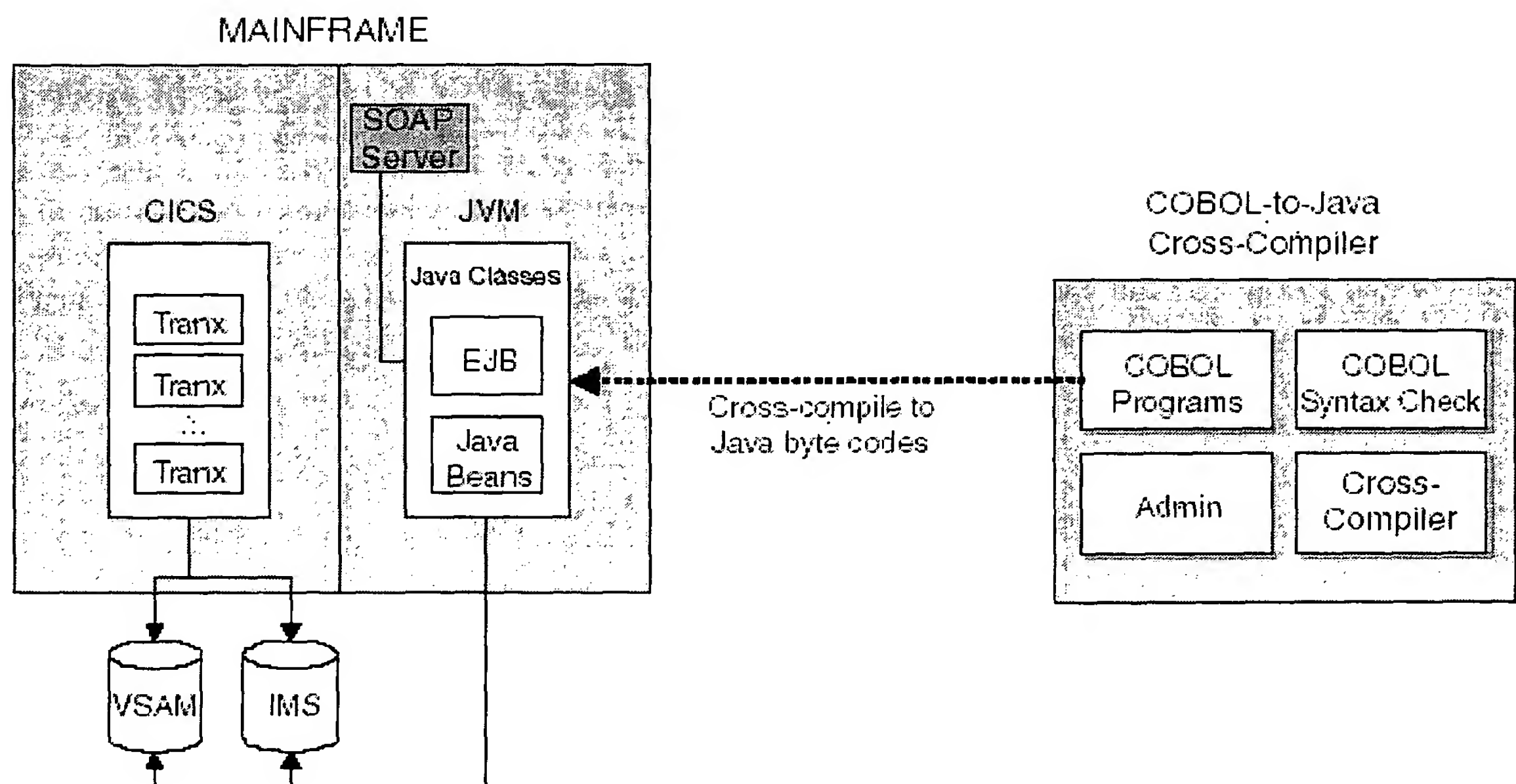


FIG. 56



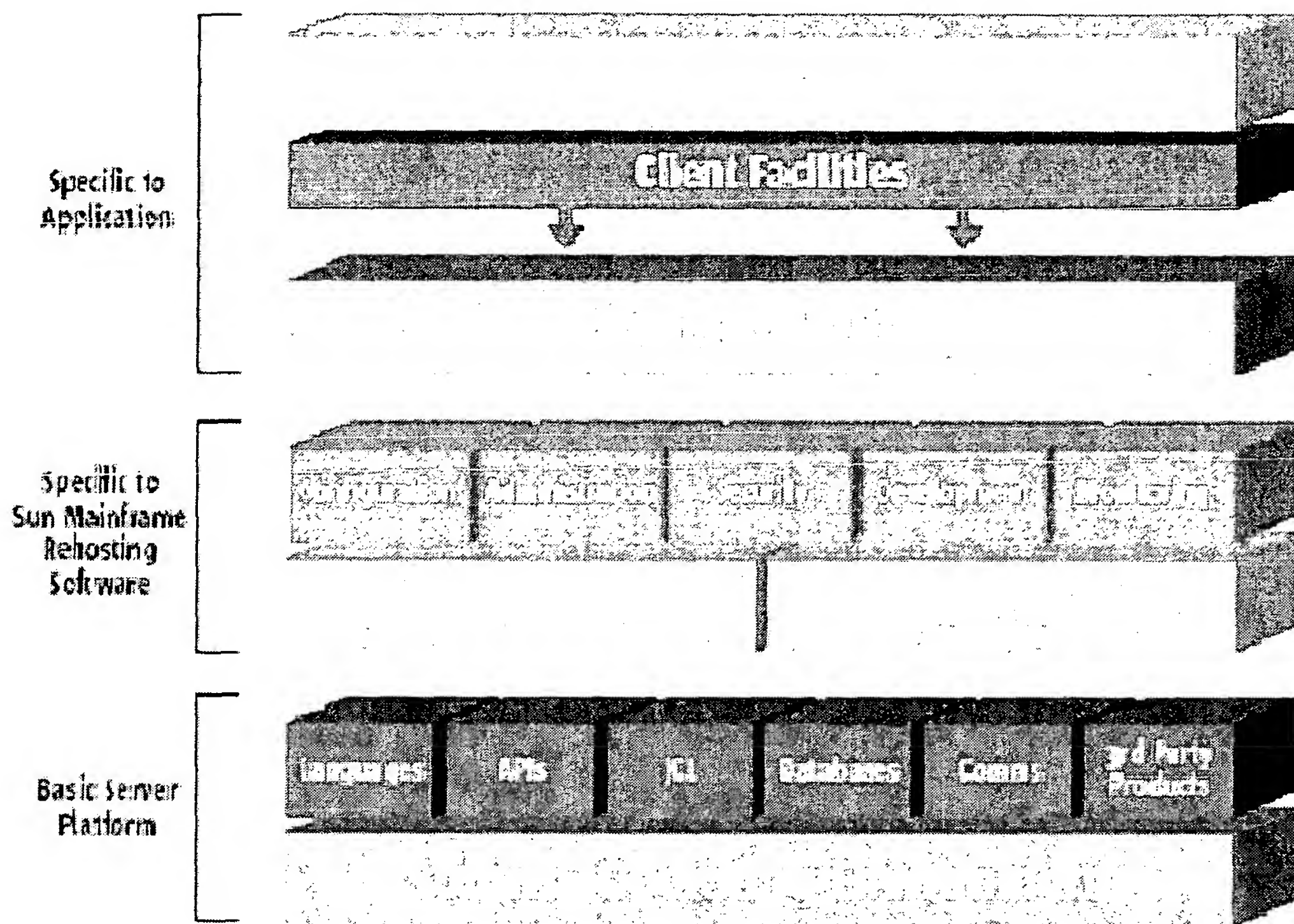


FIG. 57

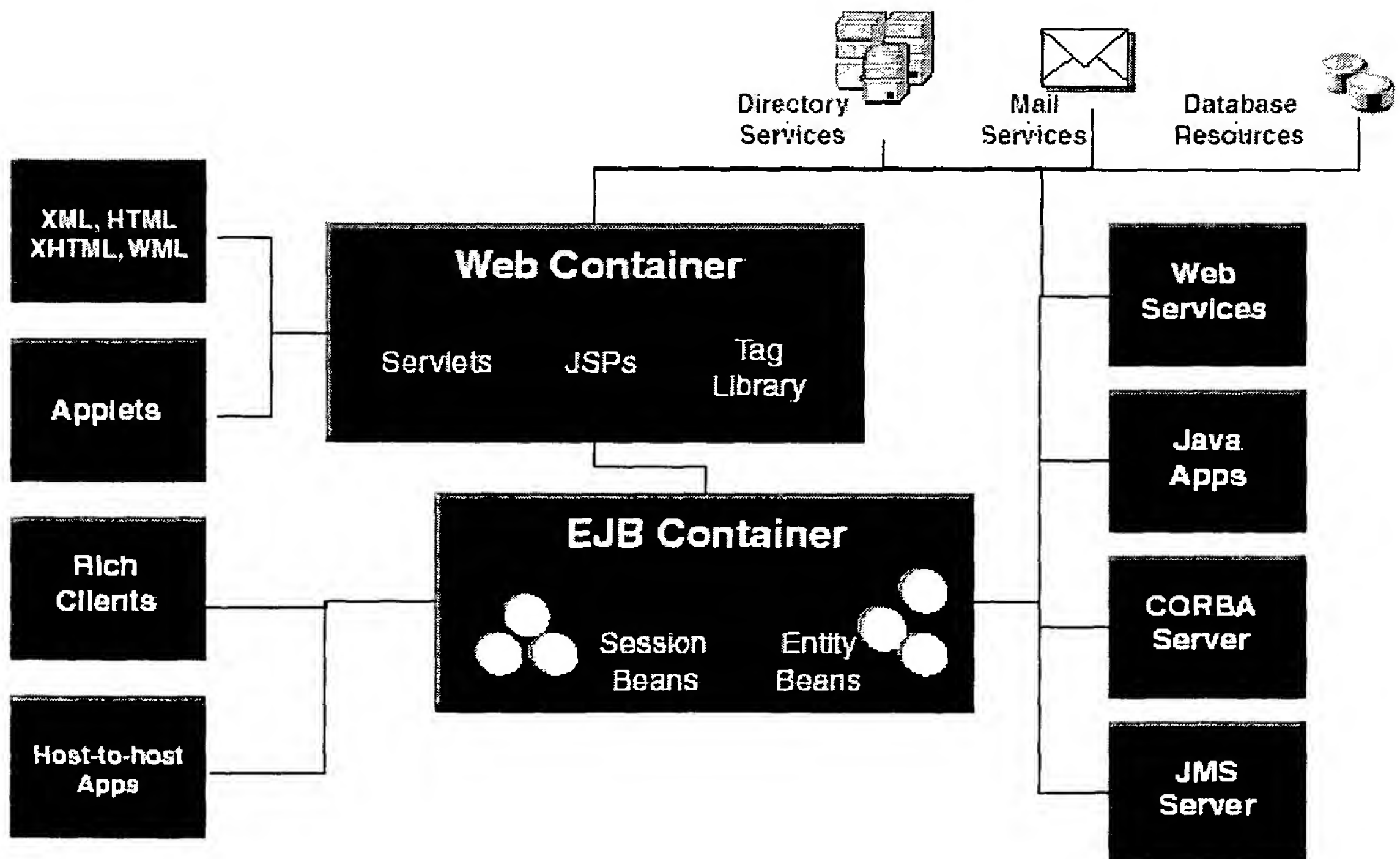
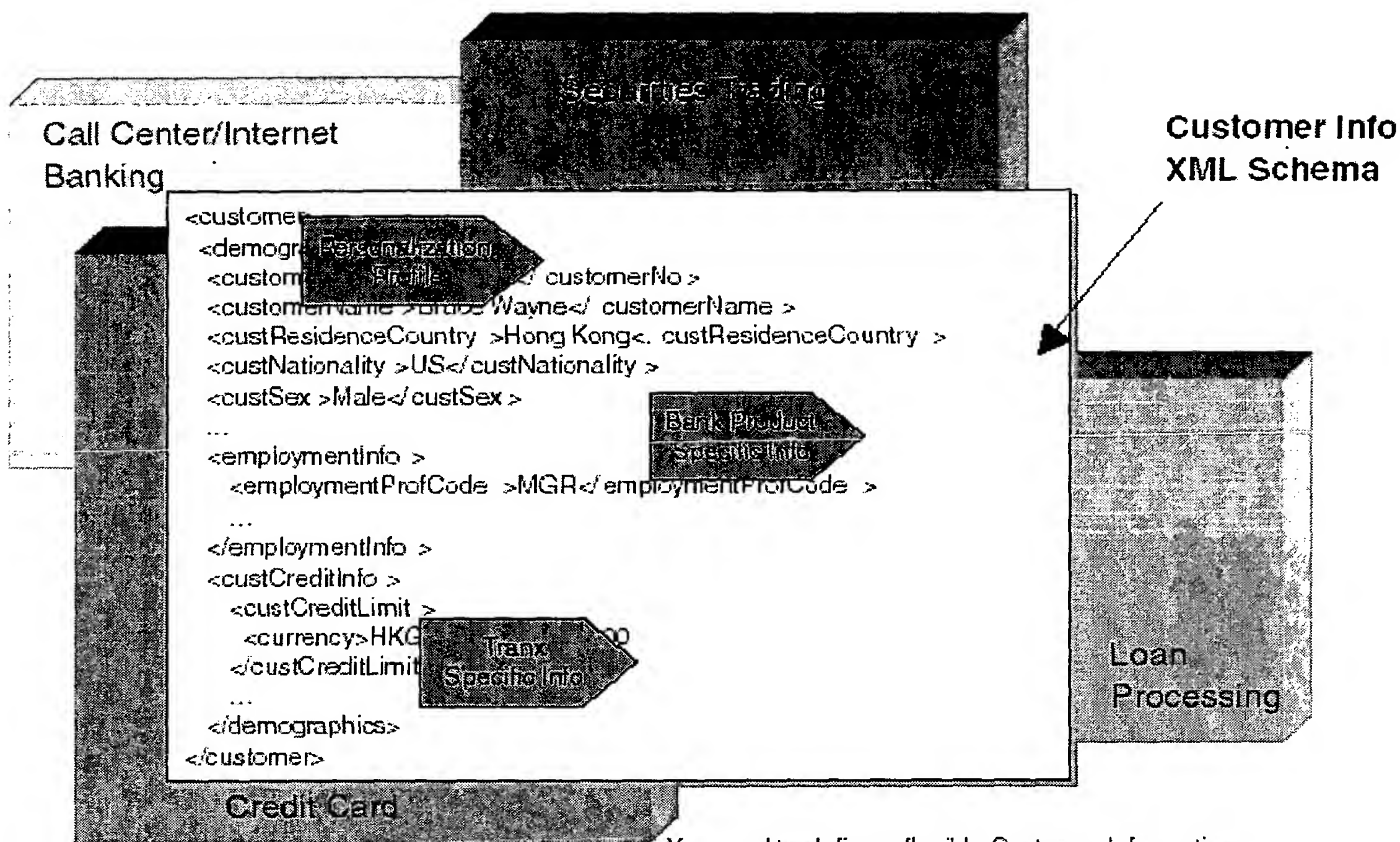


FIG. 58

Migration Approach	When to Use	Pros	Cons
Transcode	Existing legacy applications have a low complexity. This applies to both off-line and batch processing.	The legacy code conversion can be automated and thus there is a low change impact for COBOL code written in a general well-documented programming style.	There are manual changes needed for high-complexity programs with dead code.
Recompile	This is suitable for stable legacy system functionality where there is no anticipated change or no strategy for future enhancement or re-engineering.	There is minimal impact to the existing architecture. There is no need to migrate the back-end database resources.	The application requires upgrading the legacy operating system to z/OS and installing a Java Virtual Machine in an LPAR for run time. Thus, architects and developers cannot decouple the business functionality from the legacy platform.
Rehost	This applies to many batch and off-line programs.	It has a lower impact of changes.	This is not ideal for online legacy systems as this may incur considerable changes to the hardware and software environment.
Refront	This allows re-engineering of business logic incrementally.	Developers can take the chance to clean up dead code.	There is a high risk of re-engineering business logic.

**FIG. 59**



You need to define a flexible Customer Information XML schema, which can be sharable by different Banking services.

FIG. 60

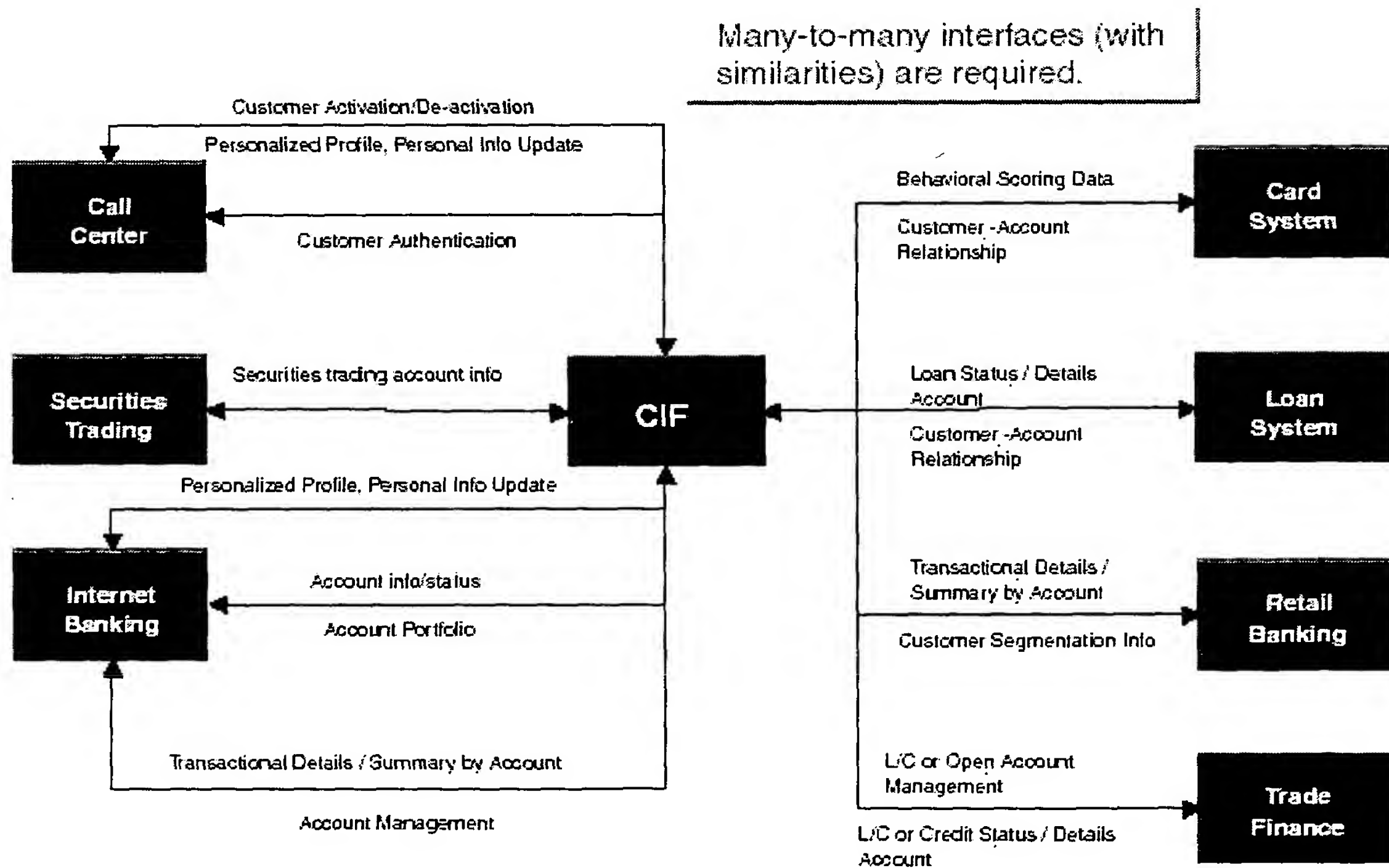


FIG. 61



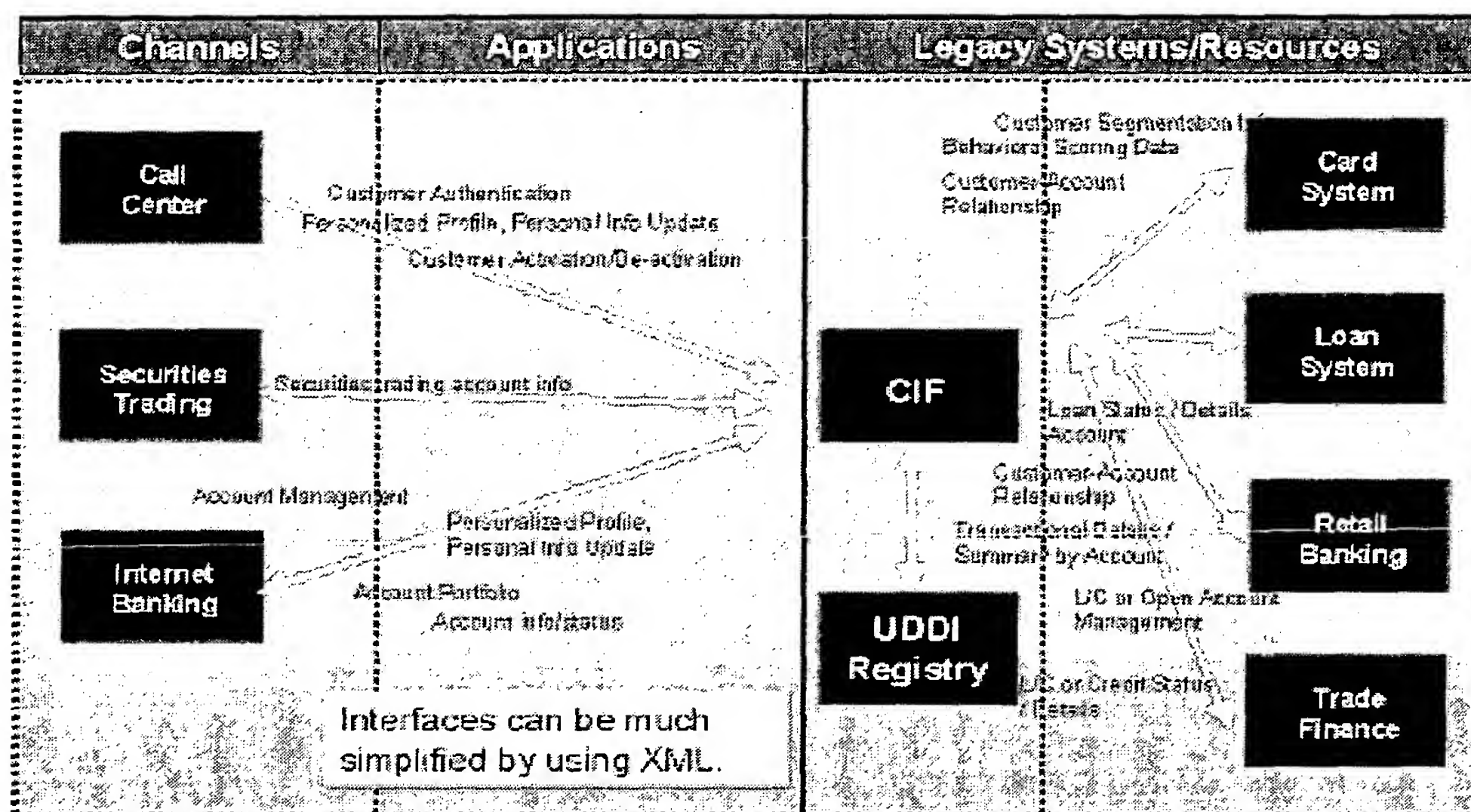


FIG. 62

### Sample Scenario—Fund Transfer

6. Complicate back-end business processes are managed by a workflow rule engine and an integration manager

5. JCA connector connects Web Services client requests to back-end legacy systems

3. Client request in SOAP messages carried over HTTP/S

1. Single Sign-on with network identity

2. Invoke authentication/entitlement services

4. Server-side SOAP component invokes legacy system functionality via XML-RPC

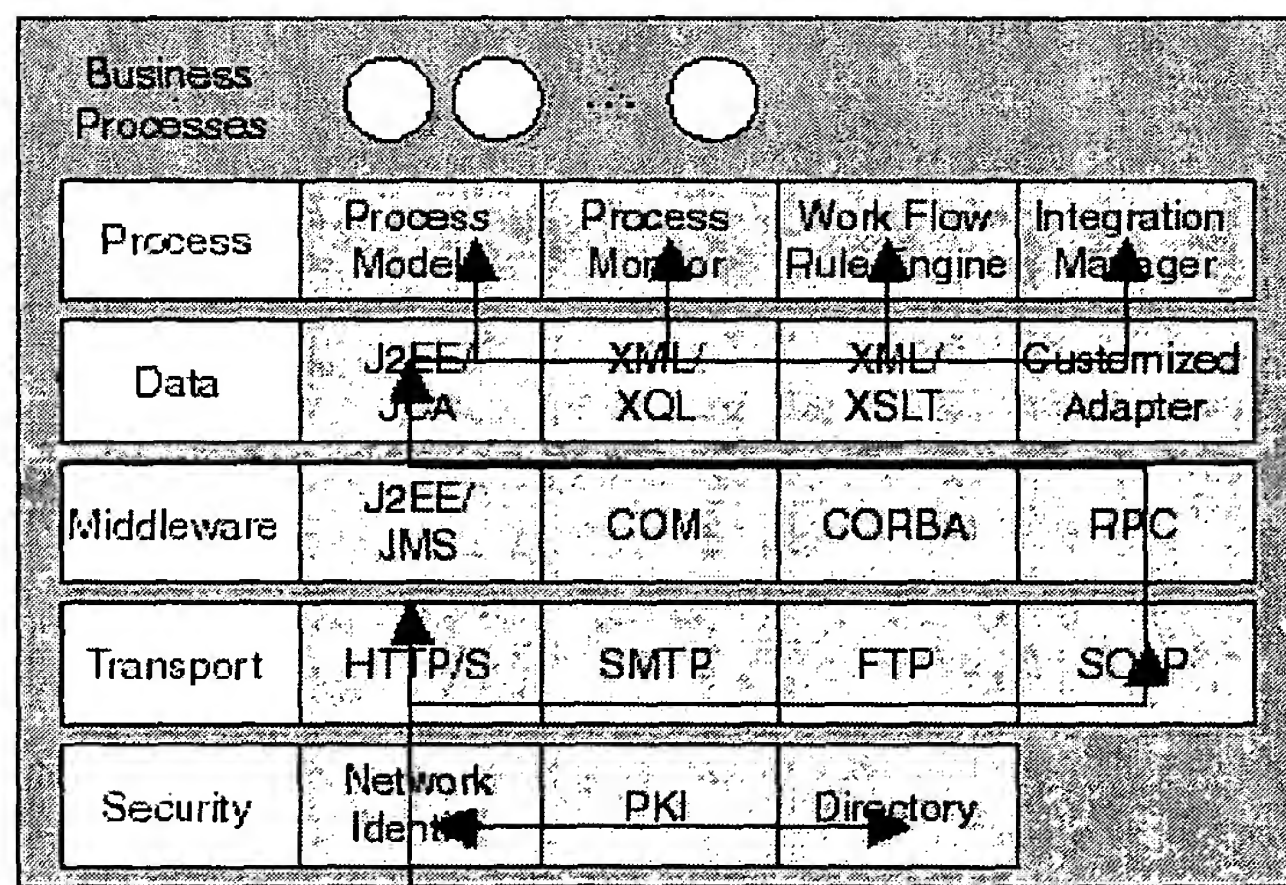


FIG. 63

	Client Tier	Presentation Tier	Business Tier	Integration Tier	Resources Tier
Application Layer				Process Models Process Monitor Workflow Rule Engine Integration Manager	
Virtual Layer		XSLT	XML	JMS RPC COM CORBA	JCA XQL
Upper Layer	HTTPS SOAP	HTTPS SOAP SMTP FTP	SOAP		
Lower Layer	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server	Network Identity/ Single Sign-on PKI Directory server

**FIG. 64**

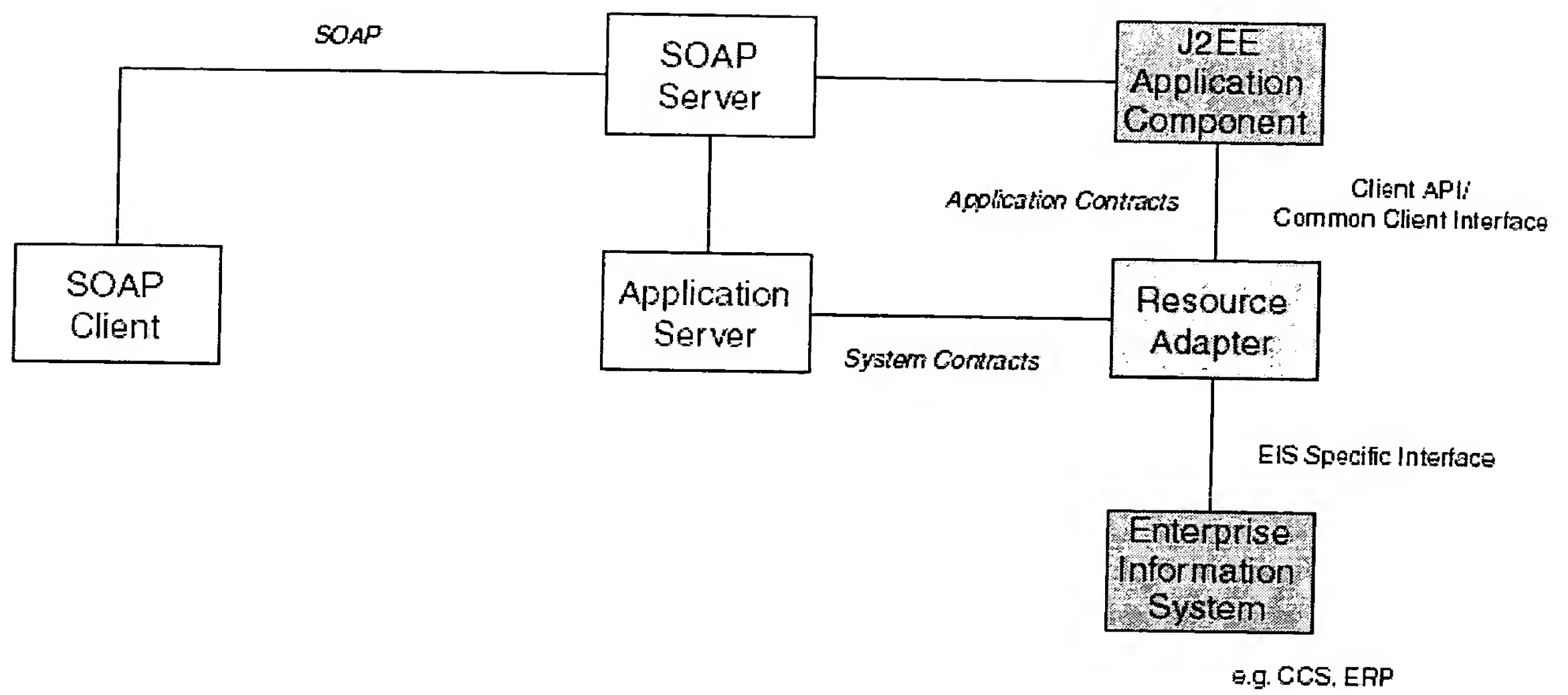


FIG. 65

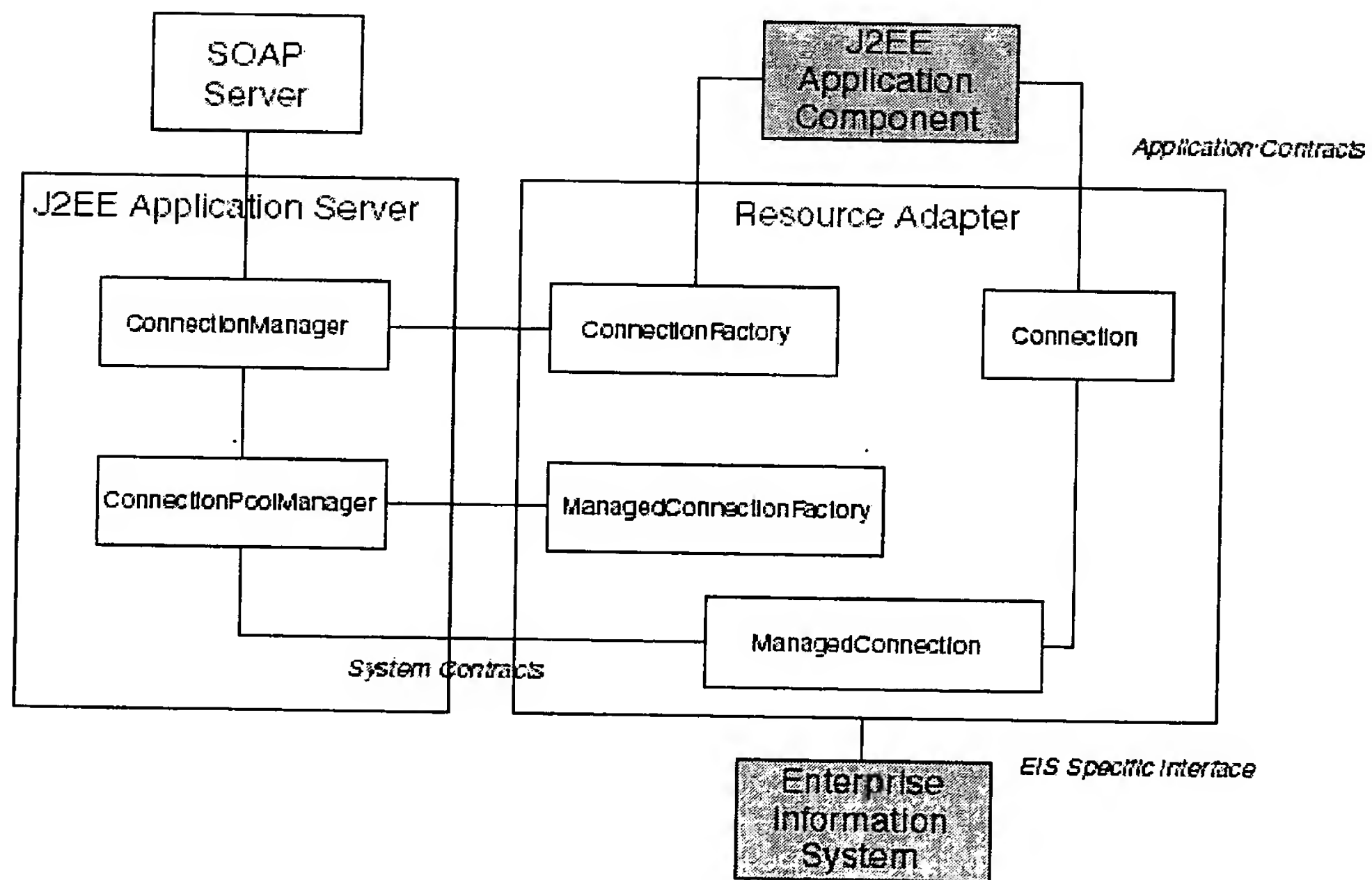


FIG. 66

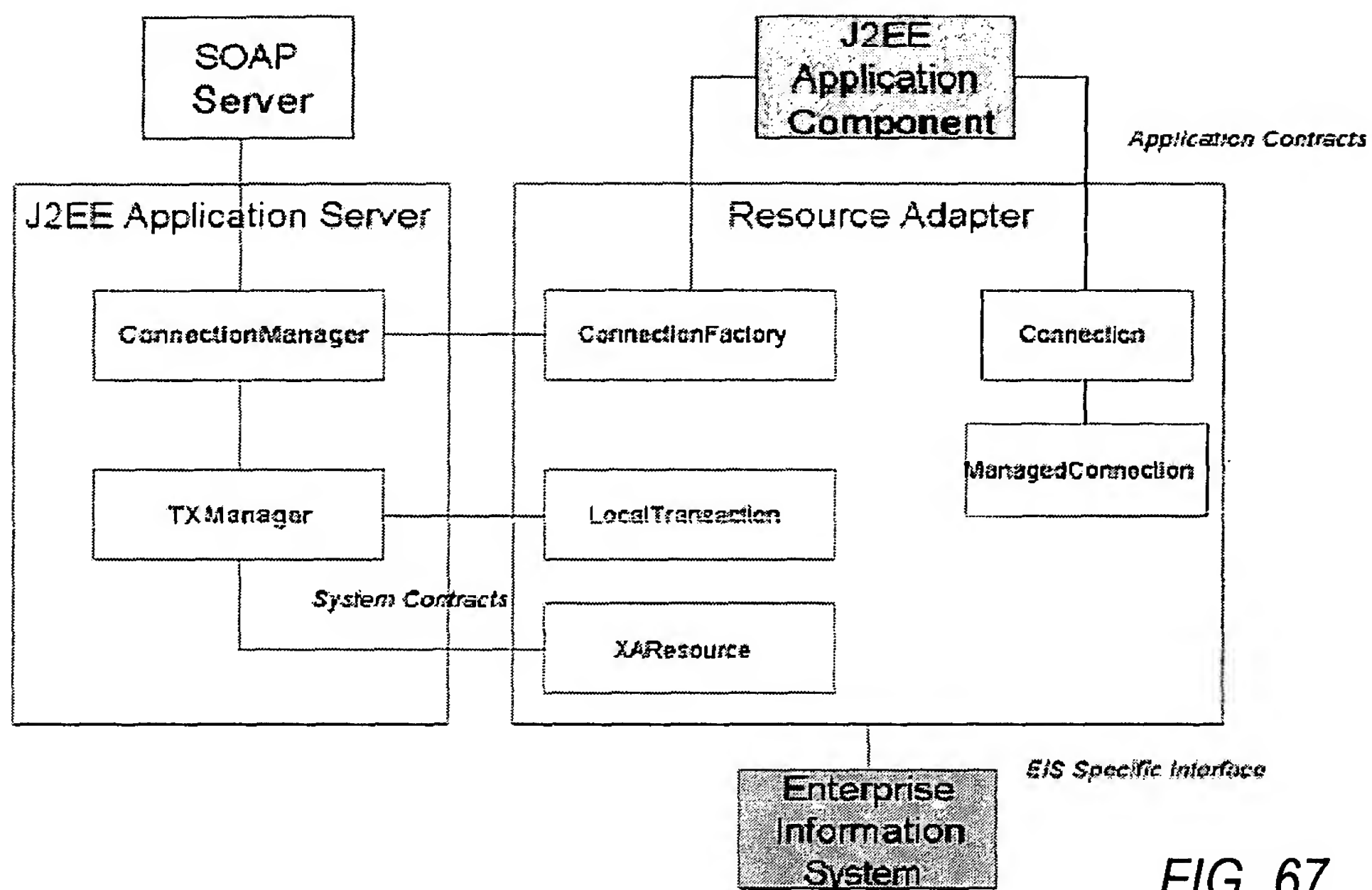


FIG. 67

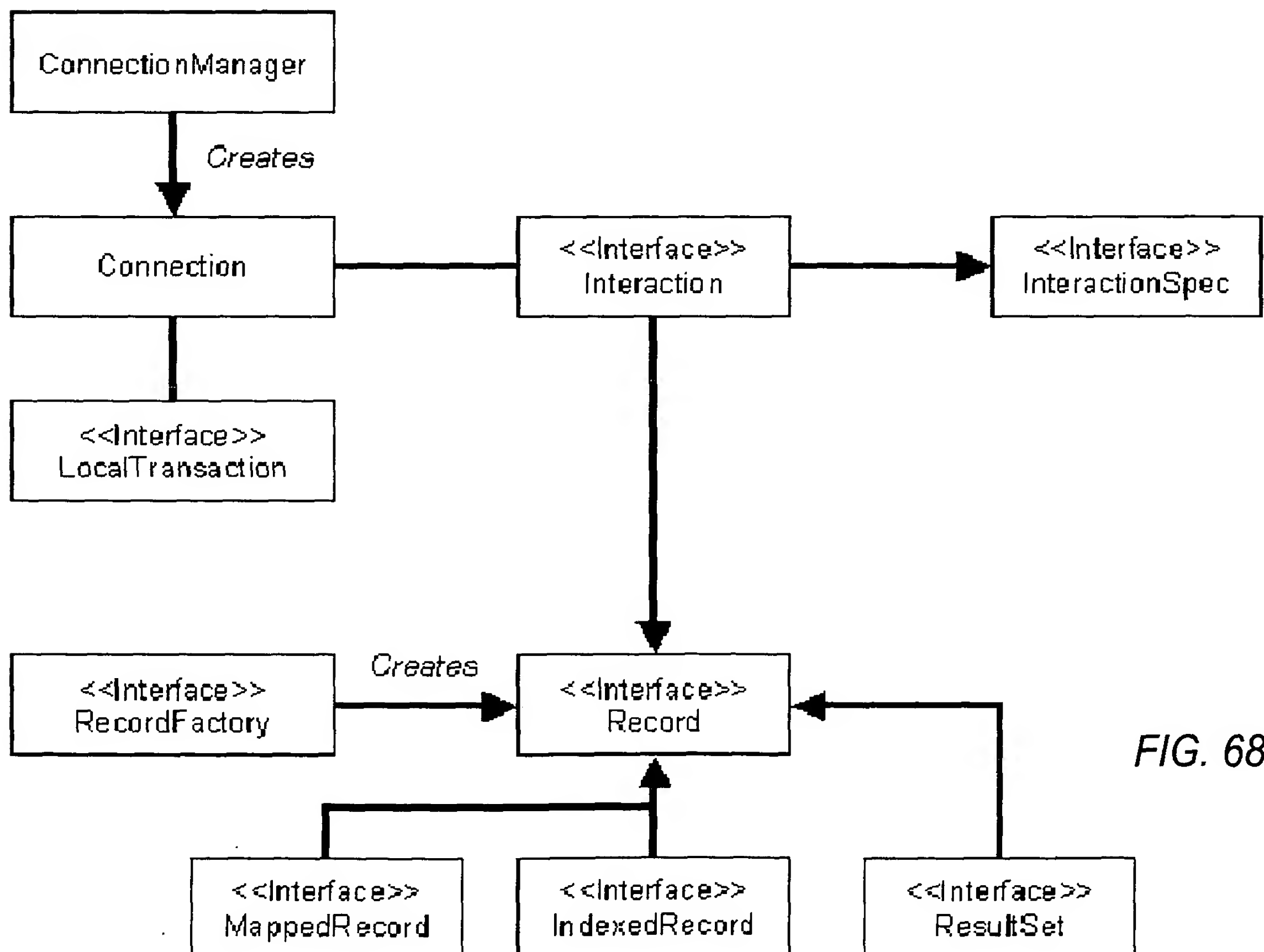


FIG. 68

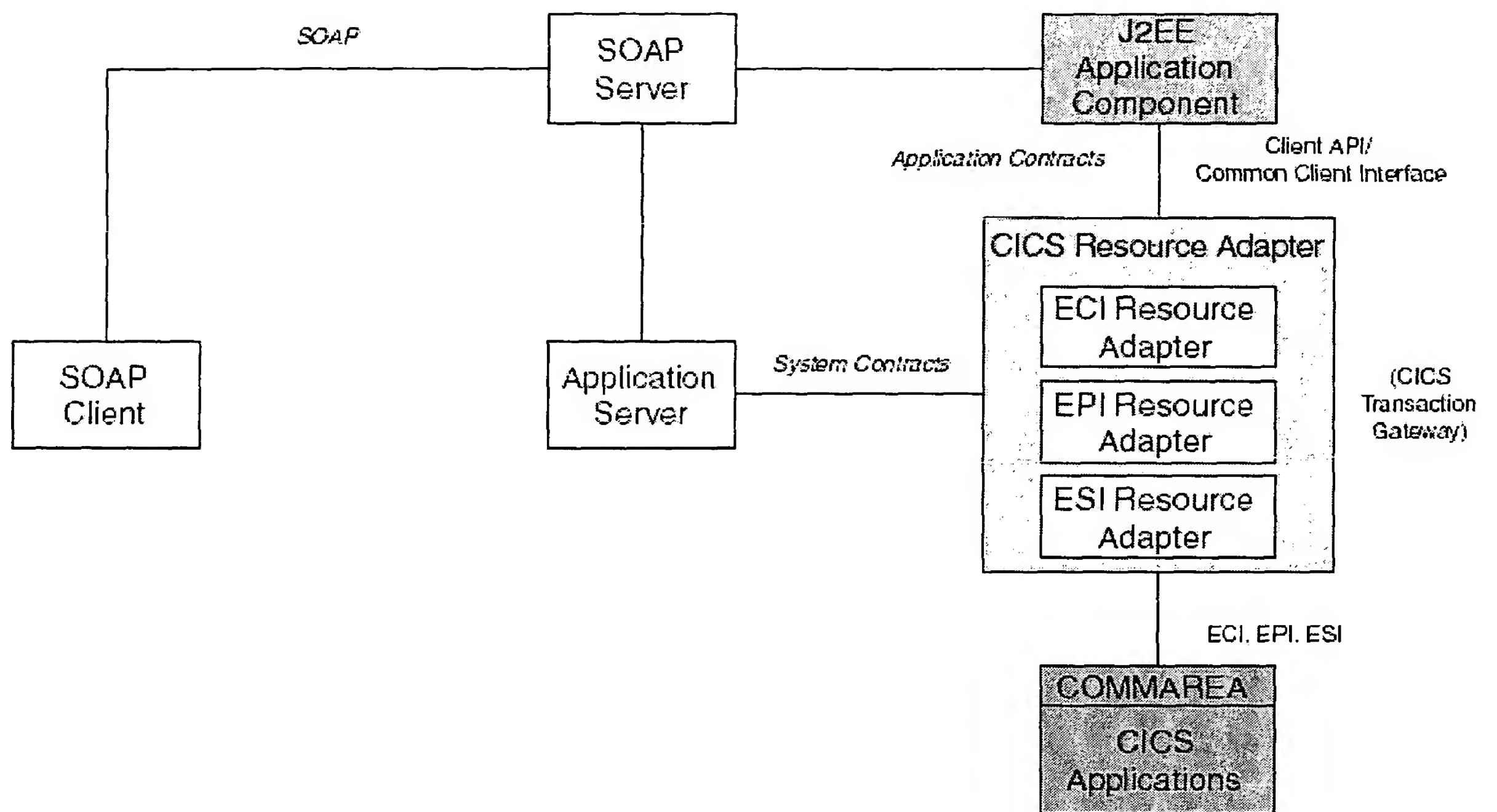


FIG. 69

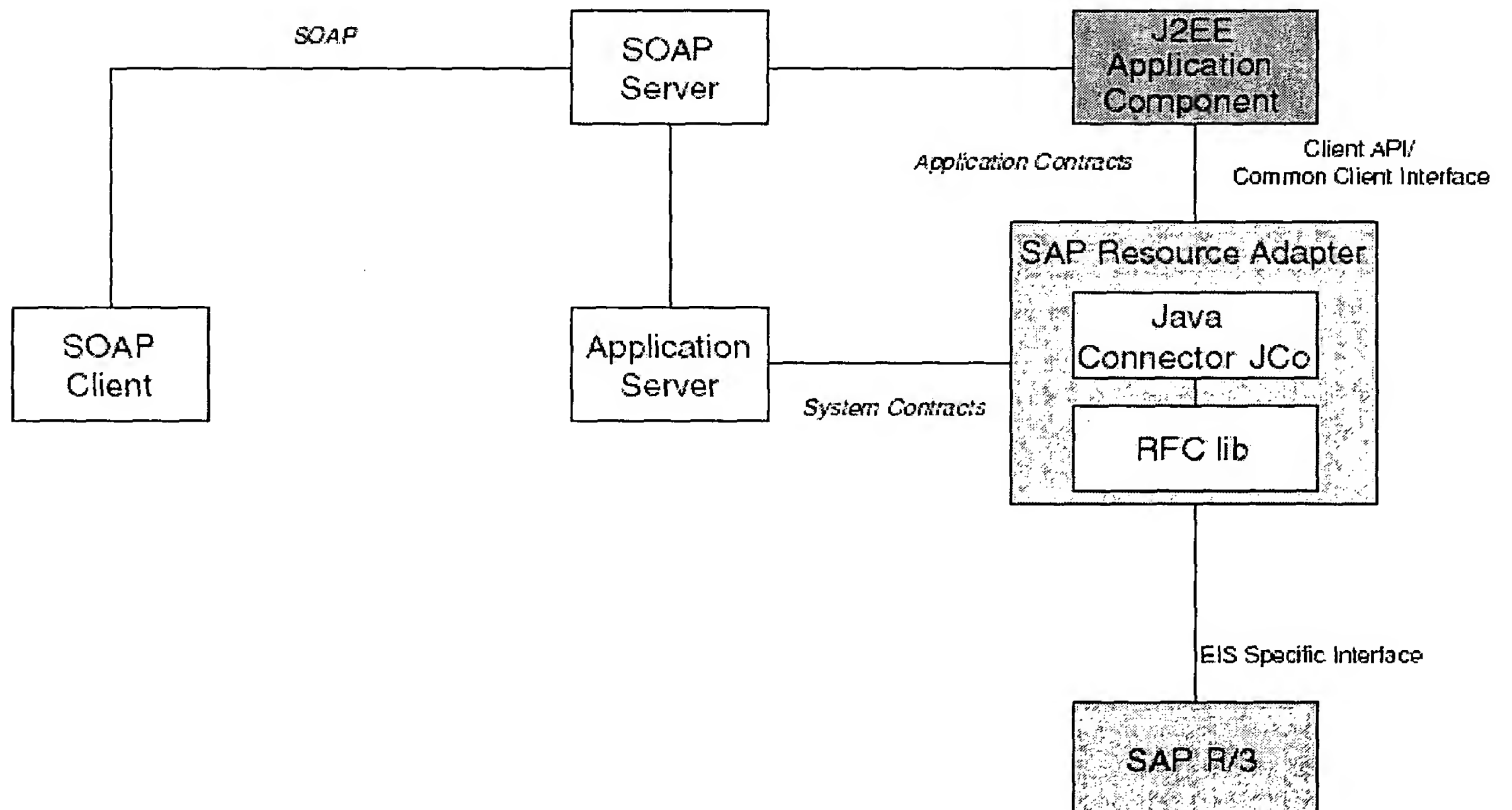
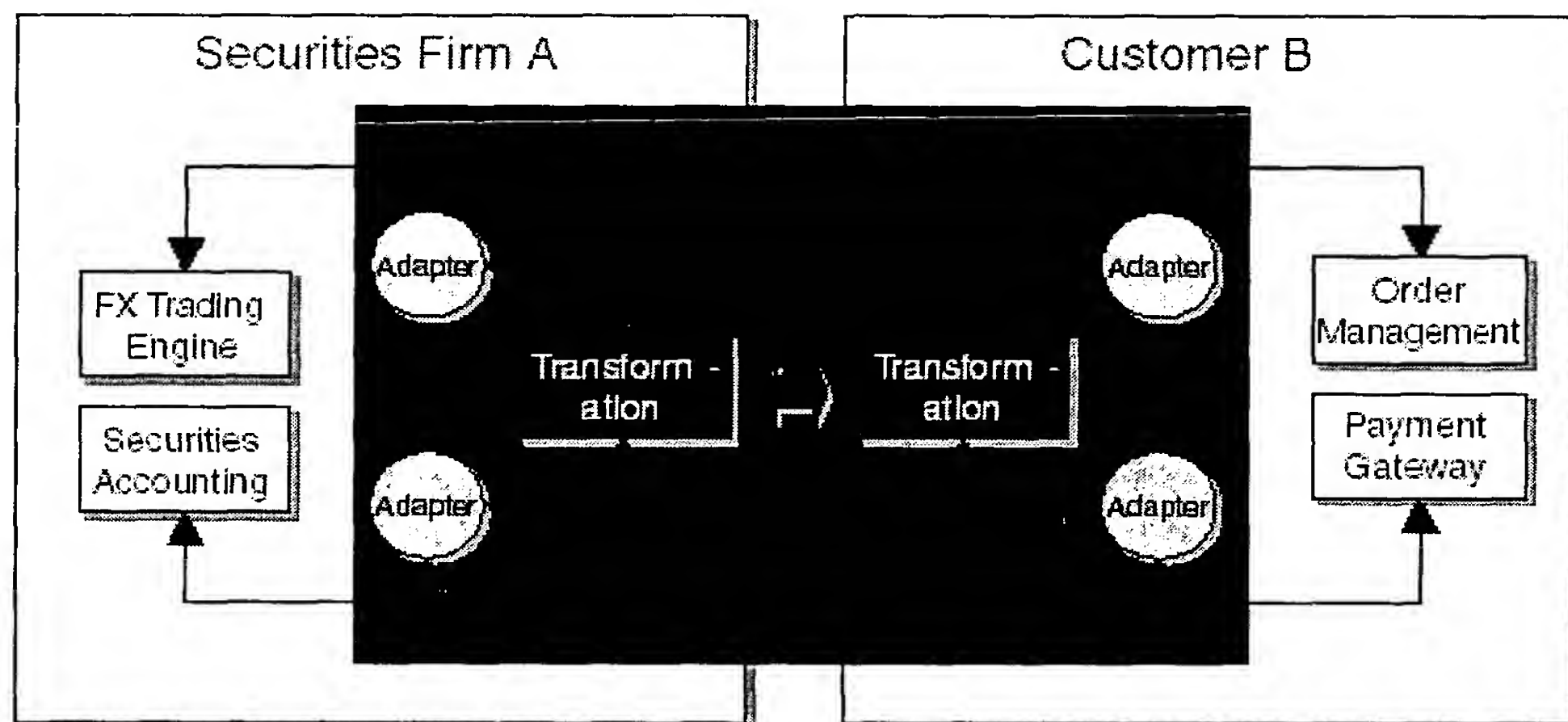


FIG. 70



Typical Technology Used  
 Customized adapter  
 Preagreed interface format/standard  
 EDI translator

When to Use  
 Point-to-point exchange, tight  
 integration  
 Limited number of trading partners  
 Relatively static data formats

Based on: Yee & Apte, Integrating Your e-Business Enterprise. SAMS, 2001.

FIG. 71



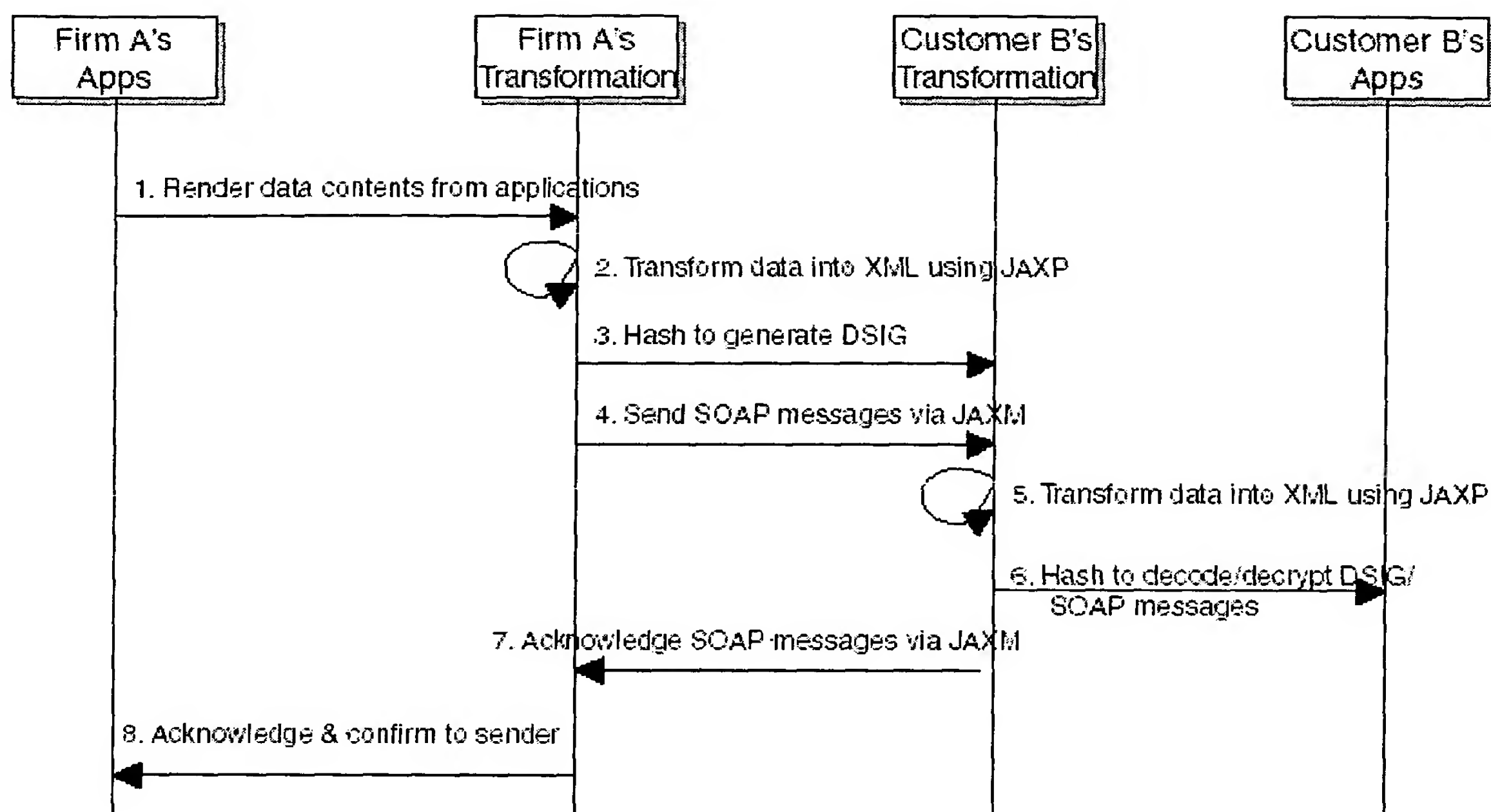
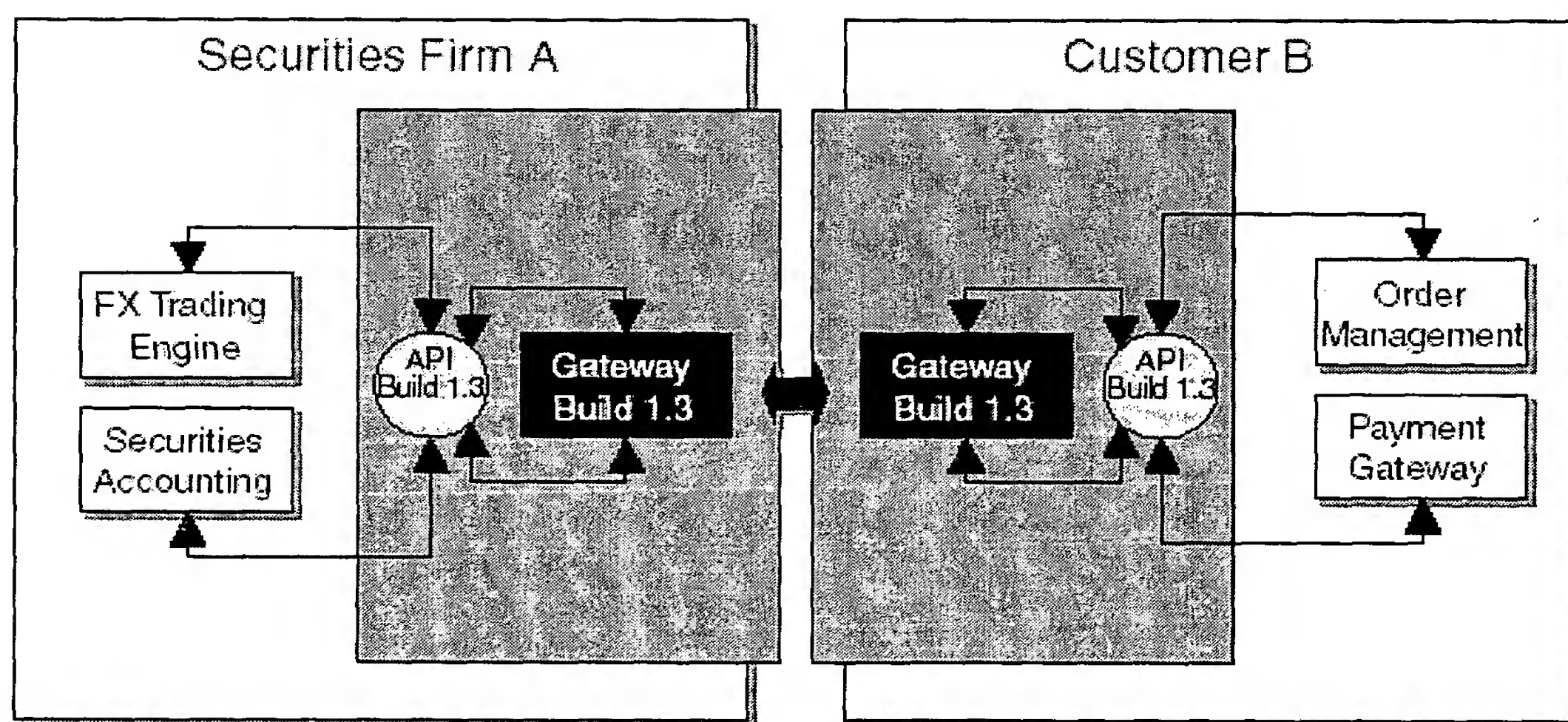


FIG. 72



Typical Technology Used  
 Standardized home-grown/customized adapter  
 Standardized interface format/API standard  
 EDI translator/EAI or middleware

When to Use  
 Strong urge for standard build  
 Point-to-point exchange, tight integration

FIG. 73

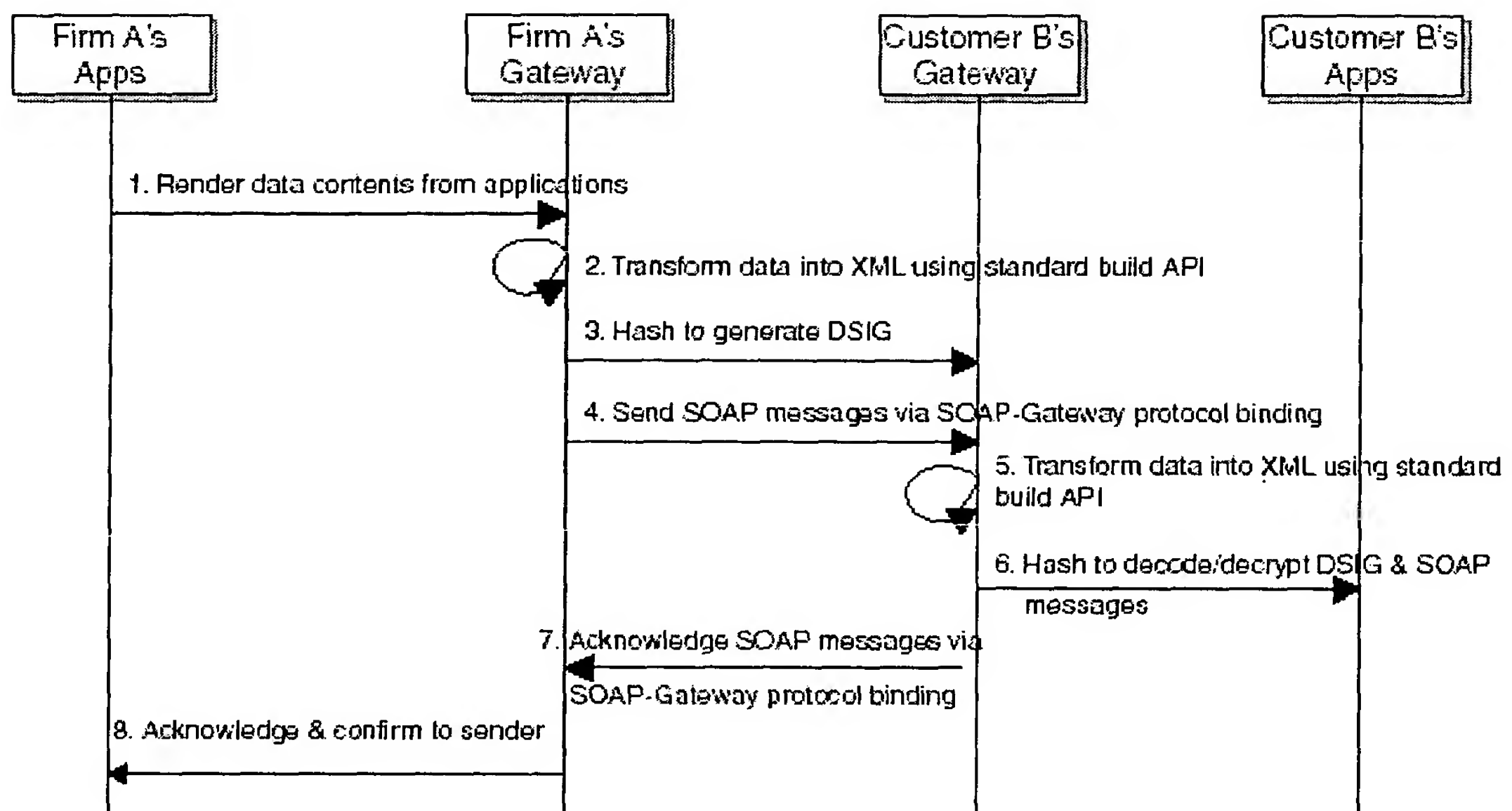
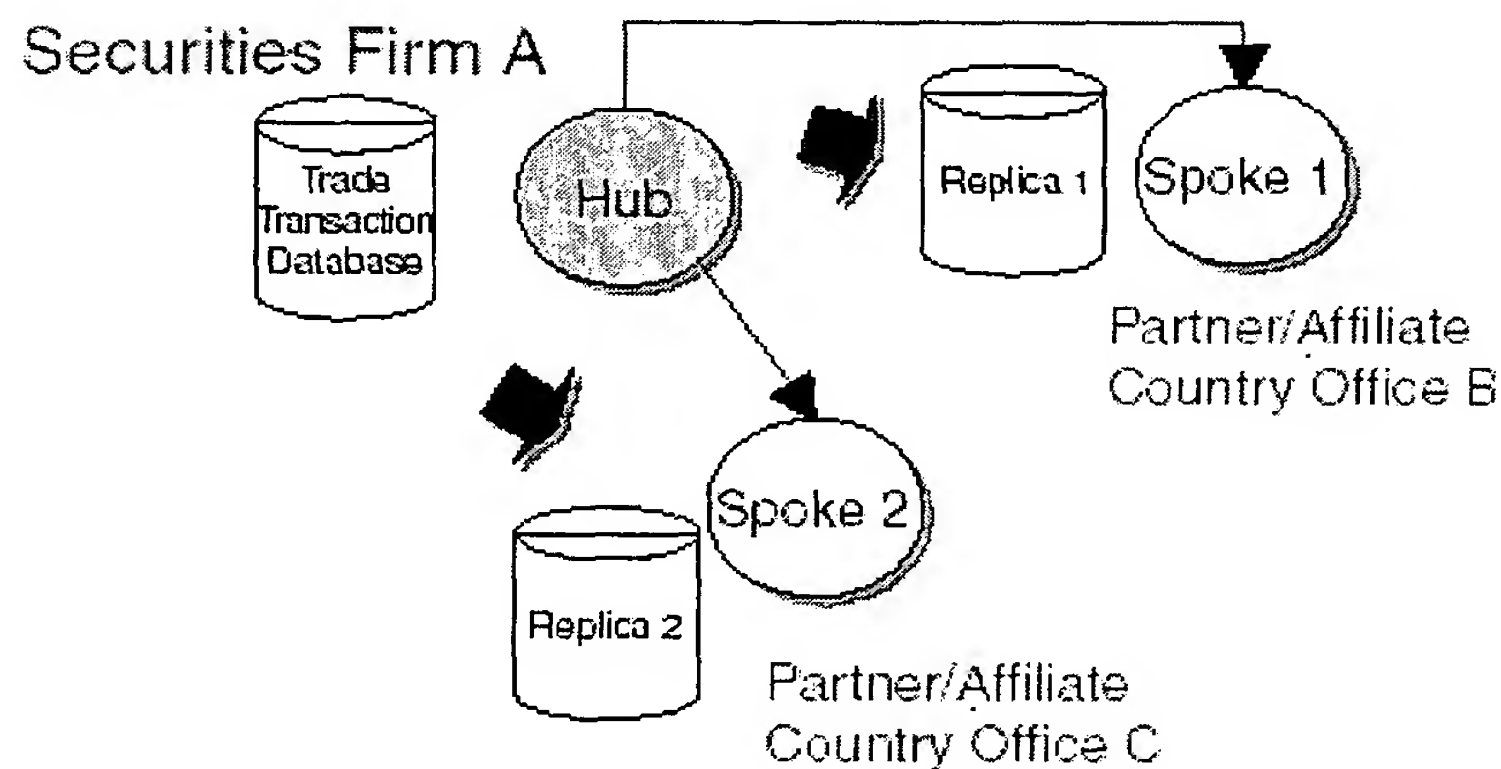


FIG. 74



Typical Technology Used  
 Synchronous/asynchronous database replication (push-pull)  
 Database/message centric applications  
 EAI/Messaging middleware (e.g., RV-TX JMS with JMS Bridge or JMS-SOAP)

When to Use  
 Highly centralized business applications  
 No geographical location constraints  
 Local spokes are for backup/performance benefits (e.g., faster access, MIS)

FIG. 75

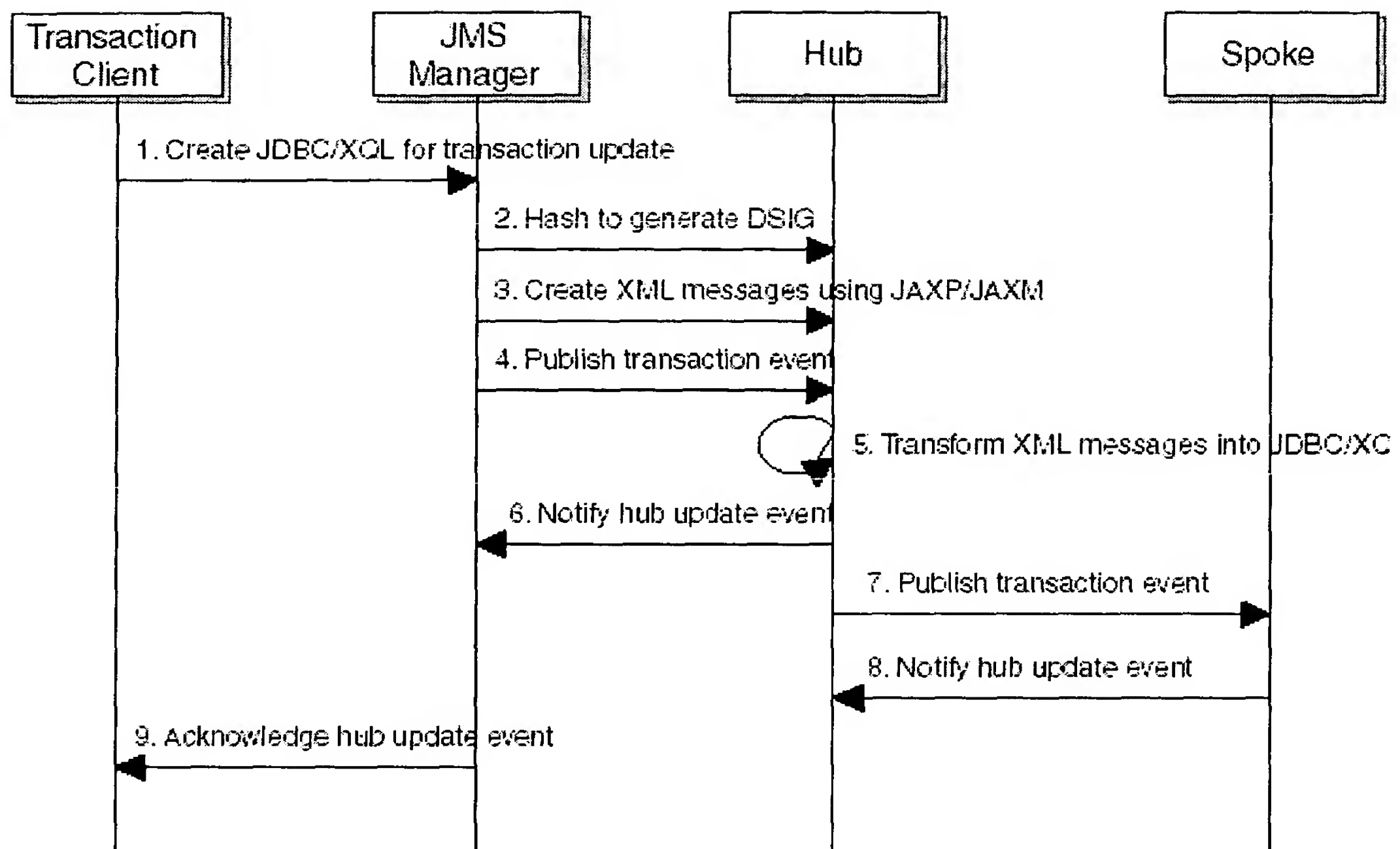
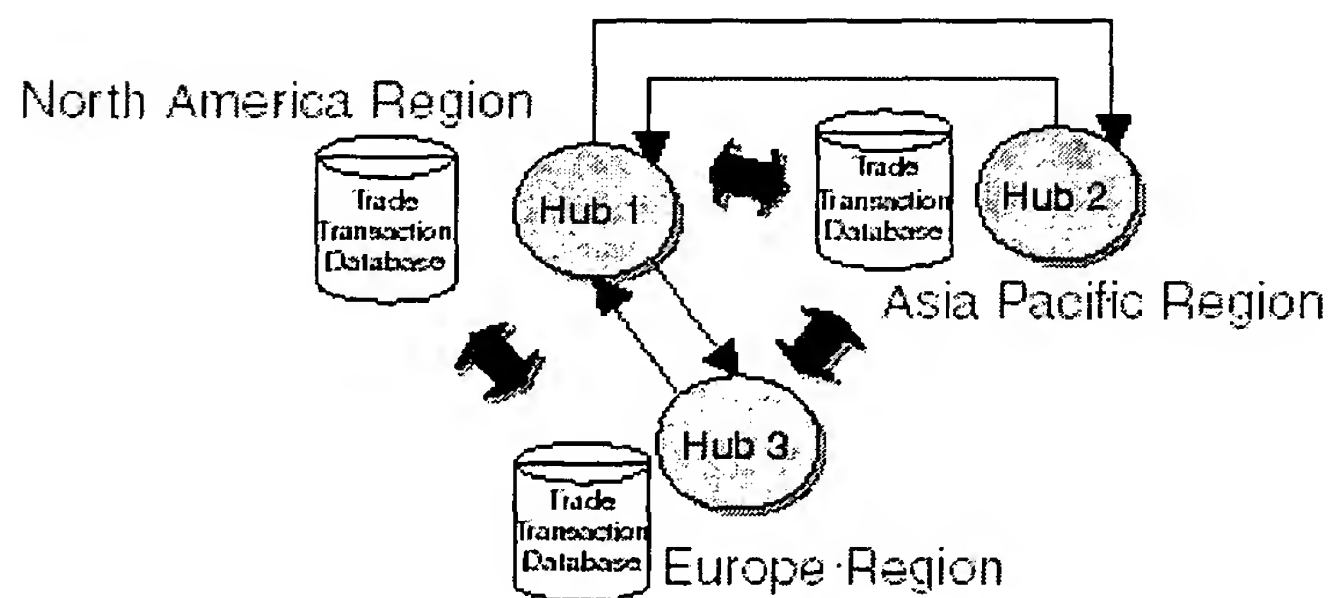


FIG. 76



Typical Technology Used  
 Synchronous/asynchronous database replication (push-push)  
 Database/Message centric applications  
 EAI/Messaging middleware, (e.g., FV-TX  
 JMS with JMS Bridge or JMS-SOAP)

When to Use  
 Highly distributed business applications with local control  
 Geographical location constraints  
 Partition different hubs for different products or transaction types, where replications are for back-up purpose

FIG. 77

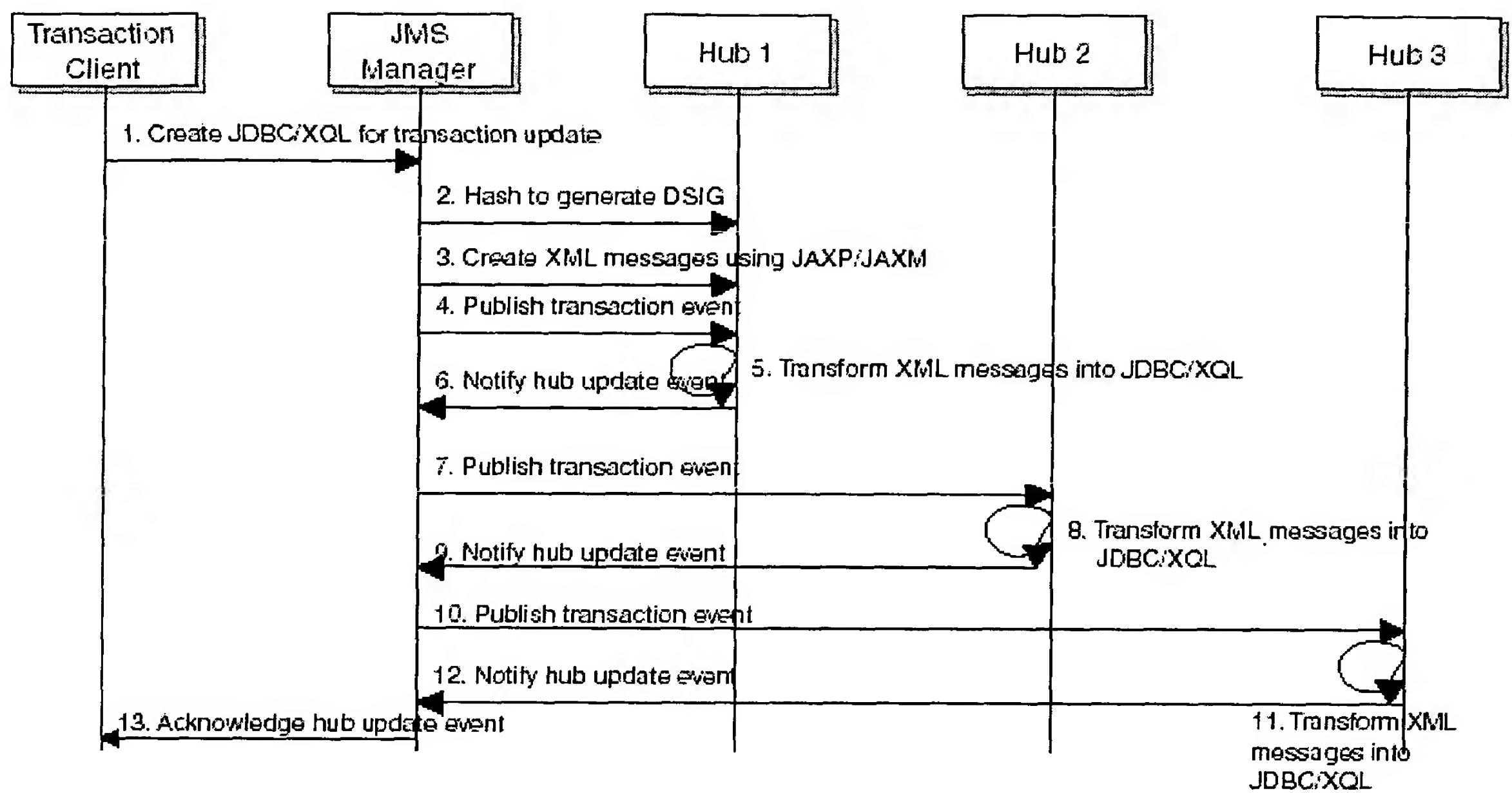


FIG. 78

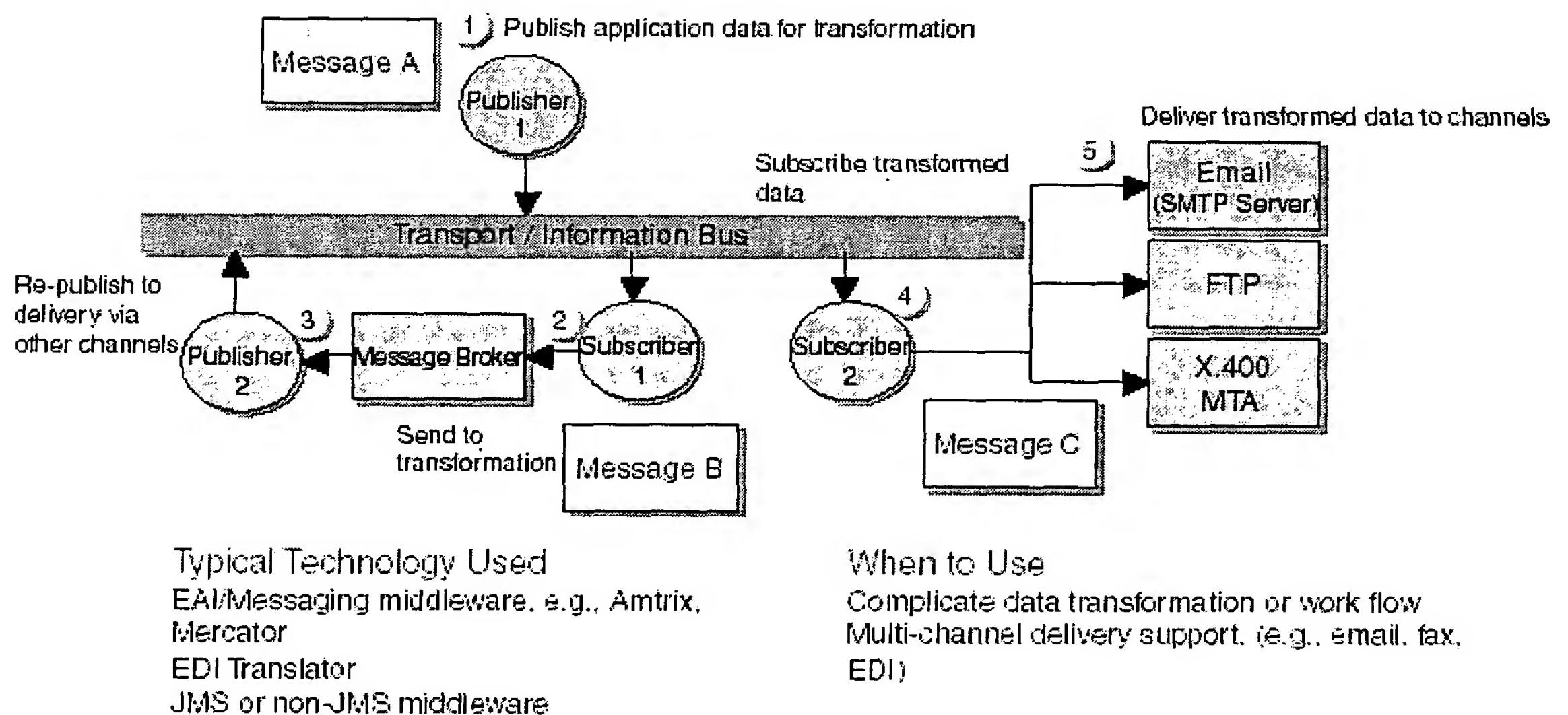


FIG. 79

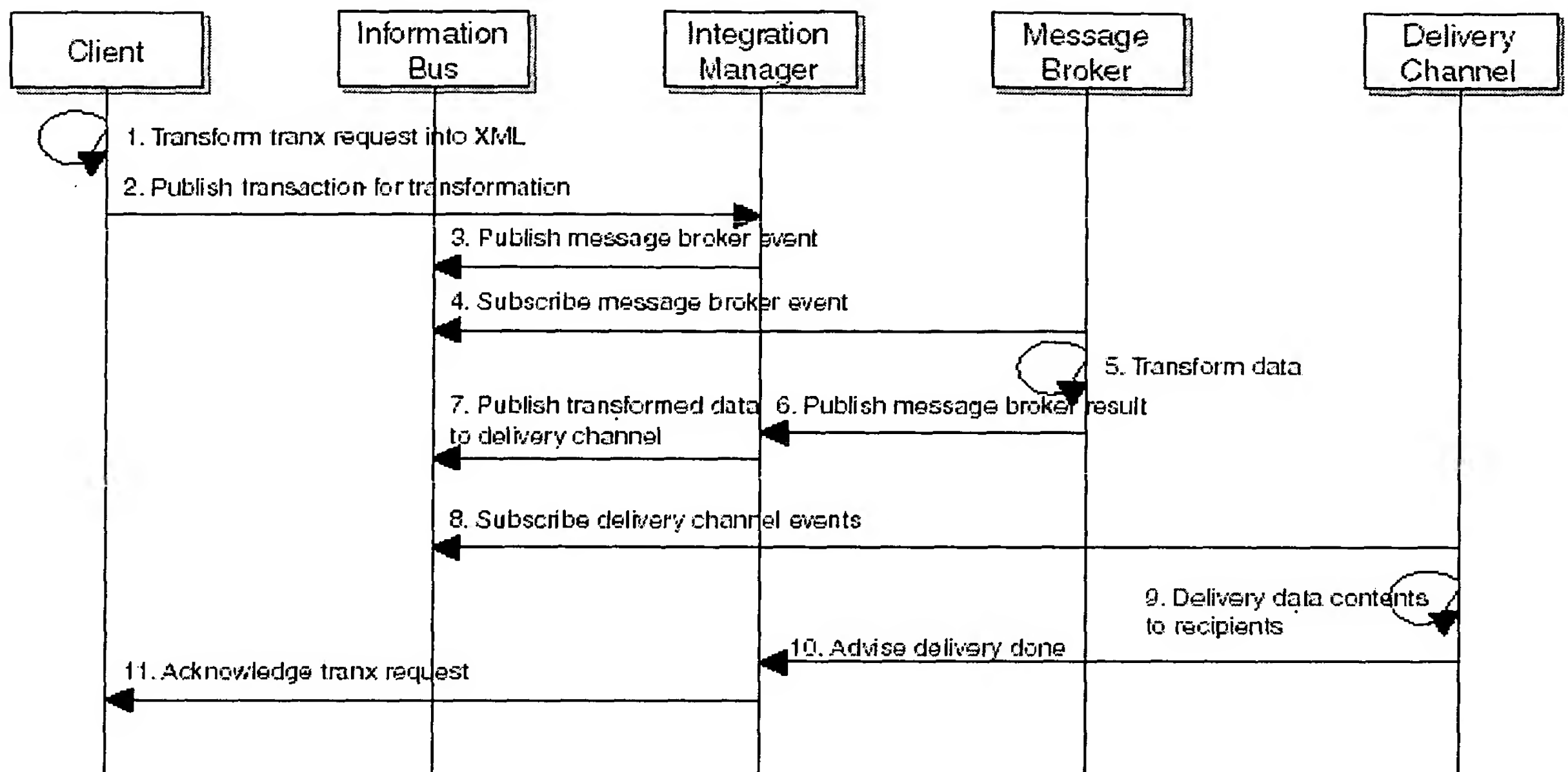
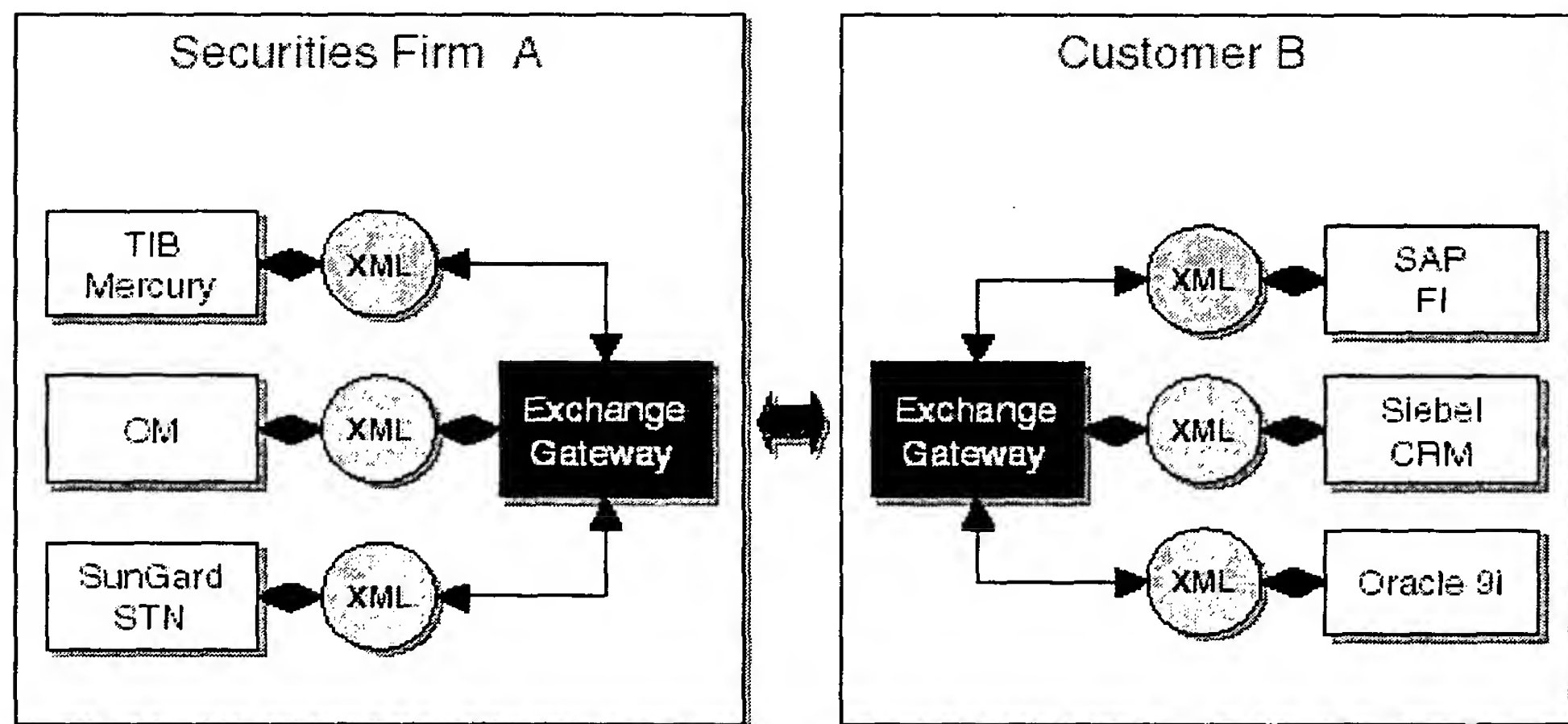


FIG. 80



Typical Technology Used  
 Vendor/off-the-shelf XML adapter  
 Preagreed XML standards/variants  
 XML Web Services

When to Use  
 Loosely coupled integration  
 Large number of trading partners  
 Multiple systems need to be integrated

Based on: Yee & Apte. Integrating Your e-Business Enterprise. SAMS, 2001.

FIG. 81

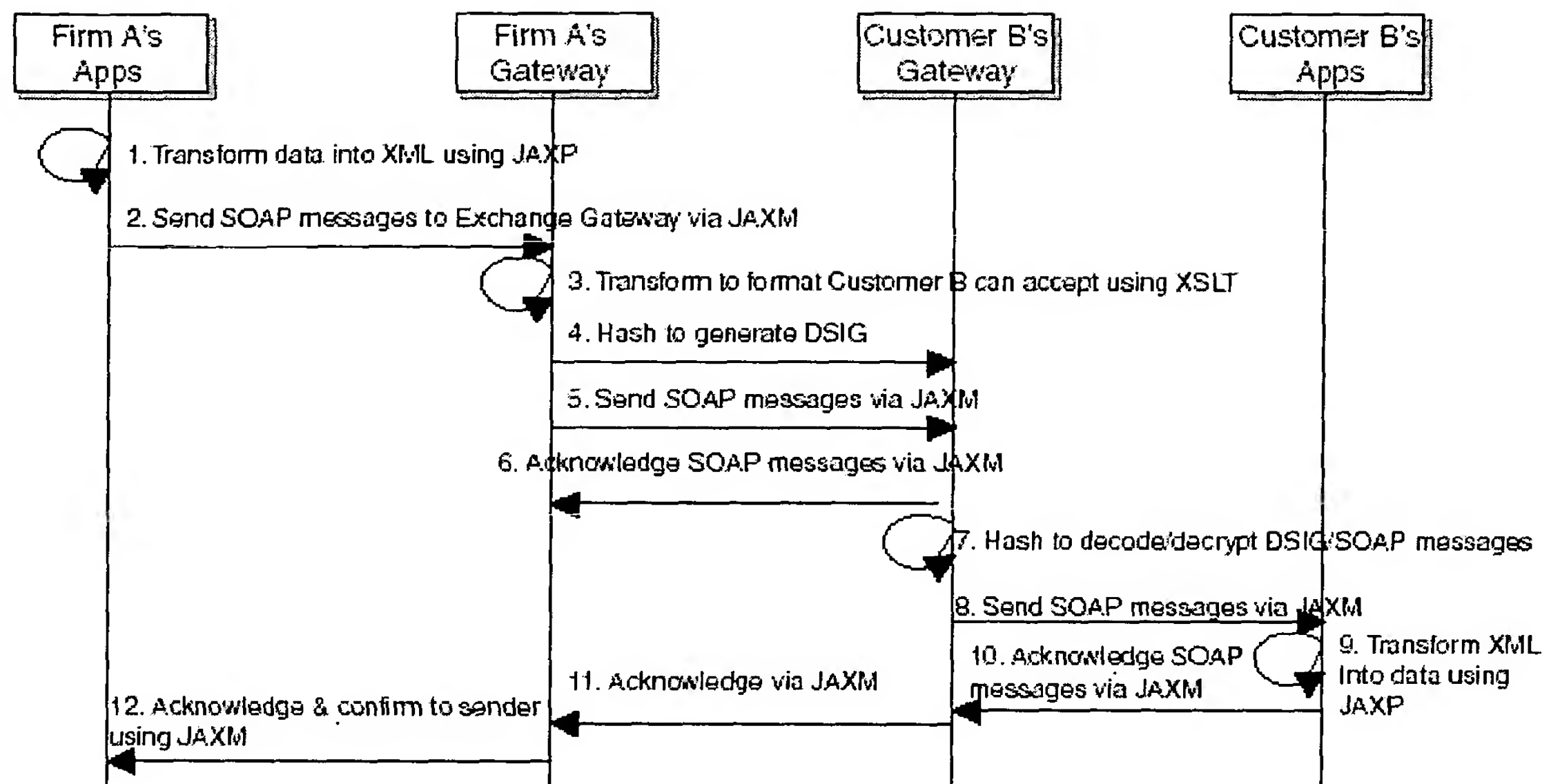
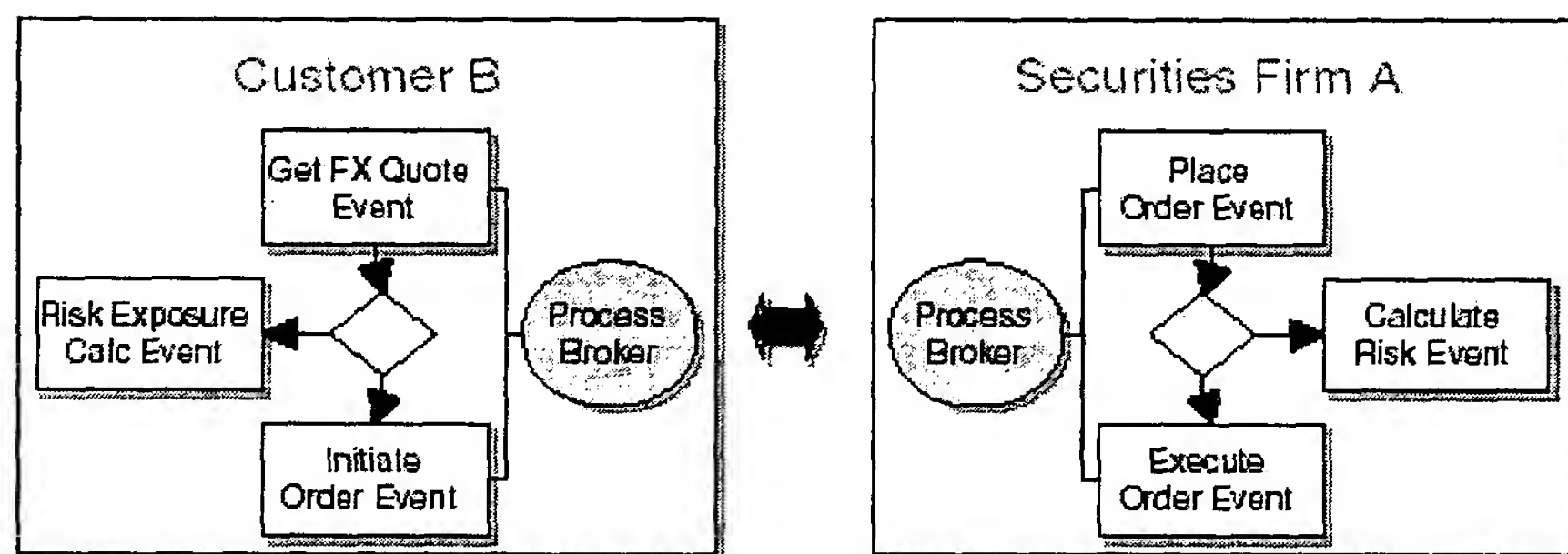


FIG. 82



Typical Technology Used  
Customized work flow integration tools  
Preagreed message formats/APIs

When to Use  
Tightly coupled integration  
Small number of trading partners  
Strong business service integration needs

Based on: Yee & Apté, Integrating Your e-Business Enterprise, SAMS, 2001.

FIG. 83



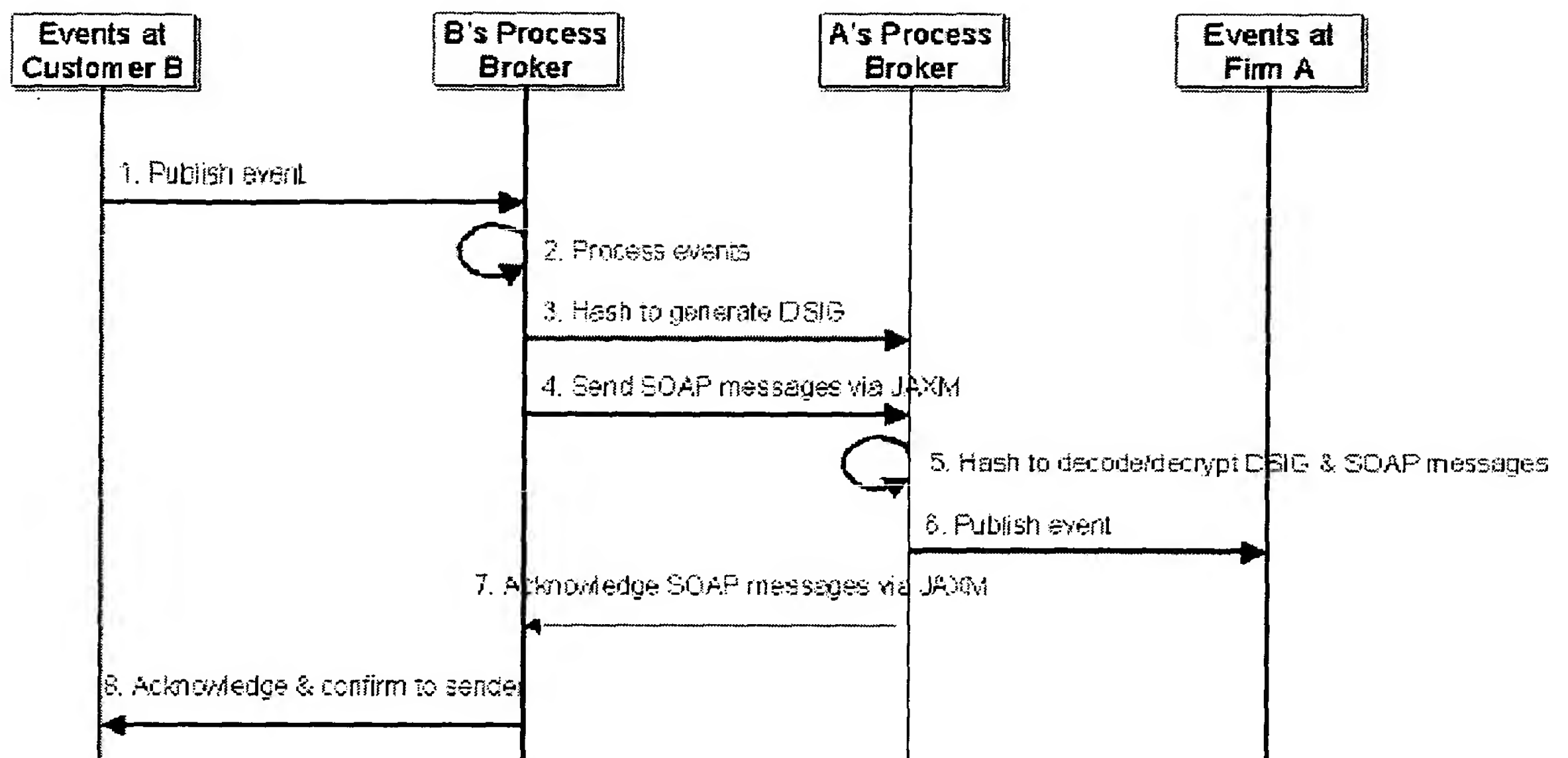
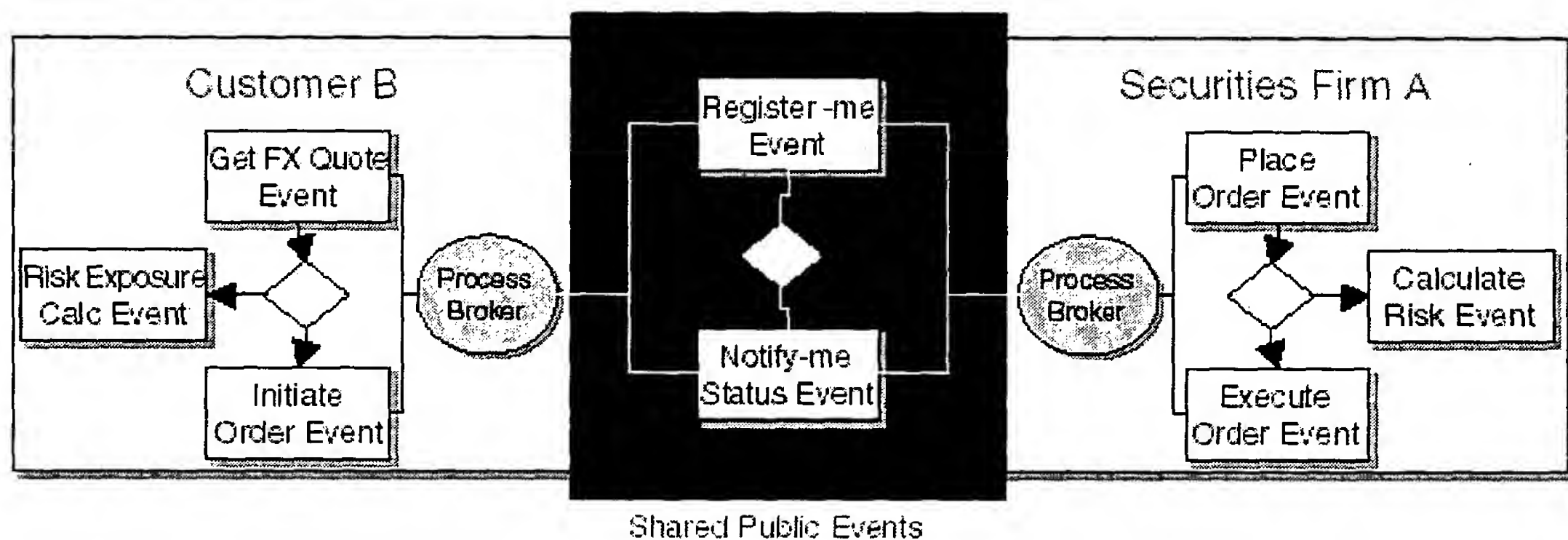


FIG. 84



Typical Technology Used  
 Customized workflow integration tools  
 Preagreed message formats/APIs  
 "Shared" process integration tools for public events

When to Use  
 "Co-branded" business services  
 Tightly coupled process & technical integration  
 Small number of trading partners

FIG. 85

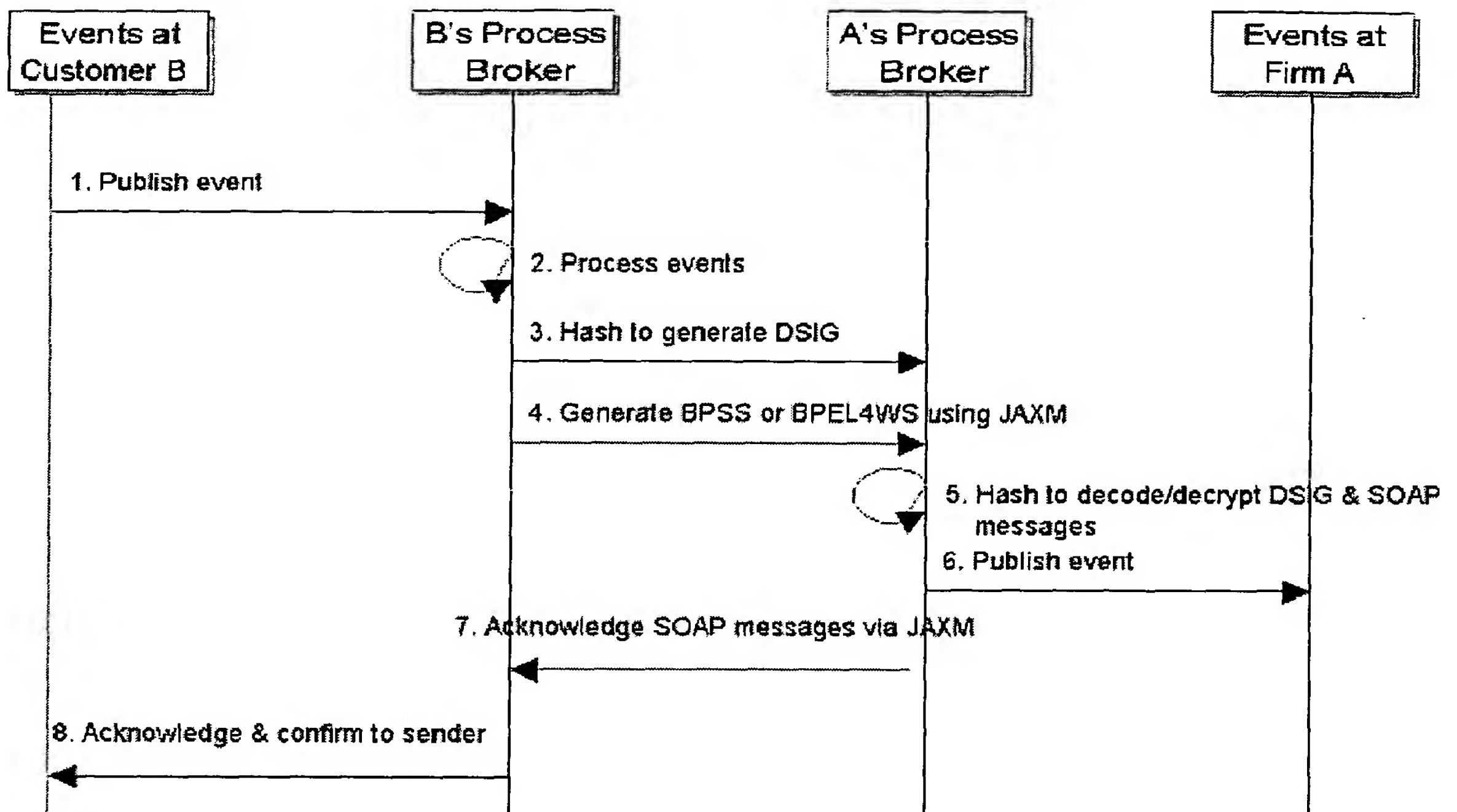
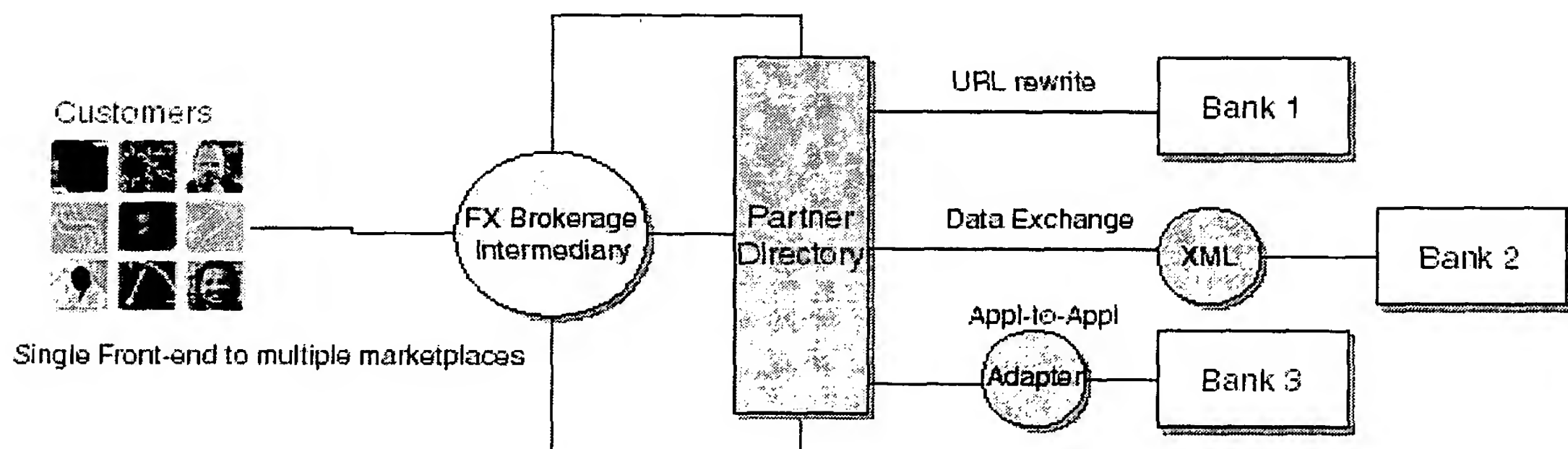


FIG. 86



Typical Technology Used  
 Hybrid integration methods  
 Pre-agreed message formats/APIs  
 XML Web Services  
 HTTP/S GET or POST

When to Use  
 Brokering similar services with a single front-end (service-provider neutral)  
 Loosely coupled process & technical integration  
 Large number of trading partners

FIG. 87

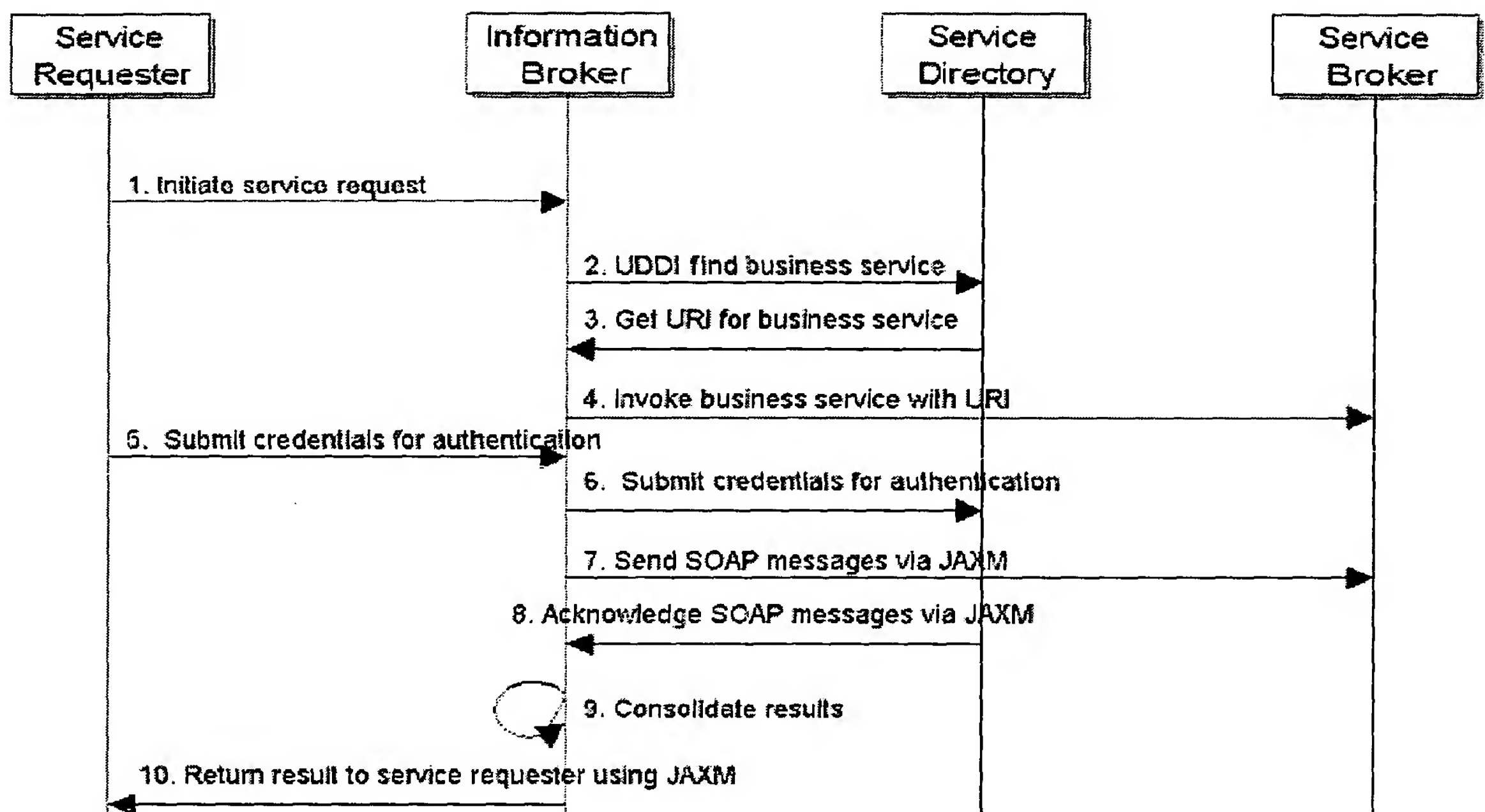
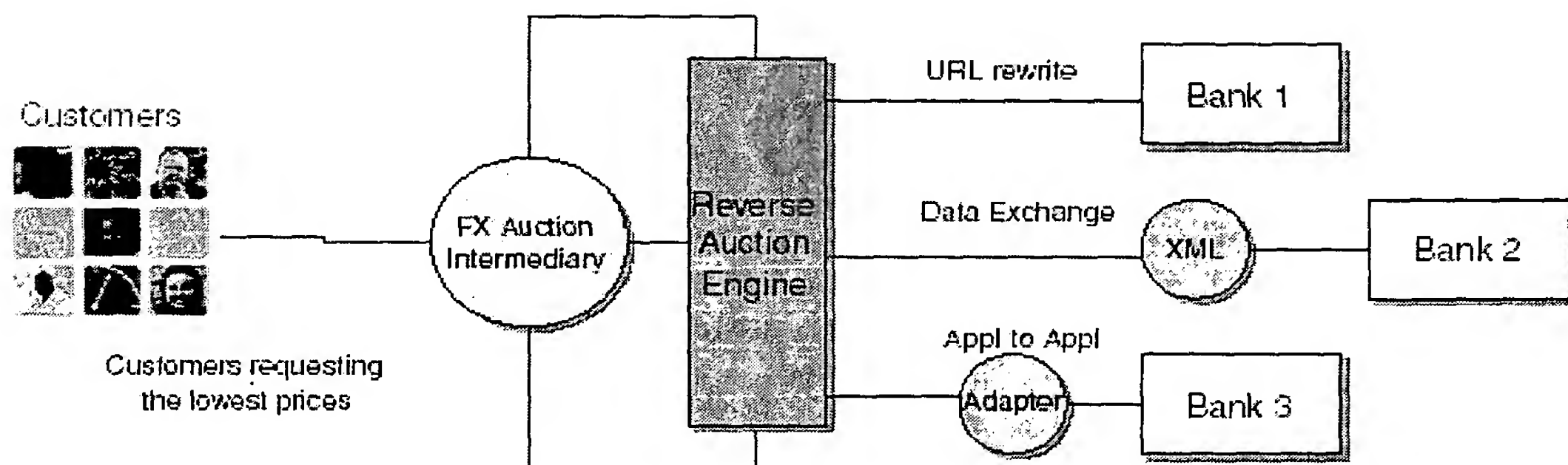


FIG. 88



Typical Technology Used  
 Hybrid integration methods  
 Preagreed message formats/APIs  
 XML Web Services  
 HTTP/S GET or POST

When to Use  
 Brokering lowest price of similar services with a single front-end (Service-Provider neutral)  
 Loosely coupled process & technical integration  
 Large number of trading partners  
 Price-sensitive & homogeneous products

FIG. 89

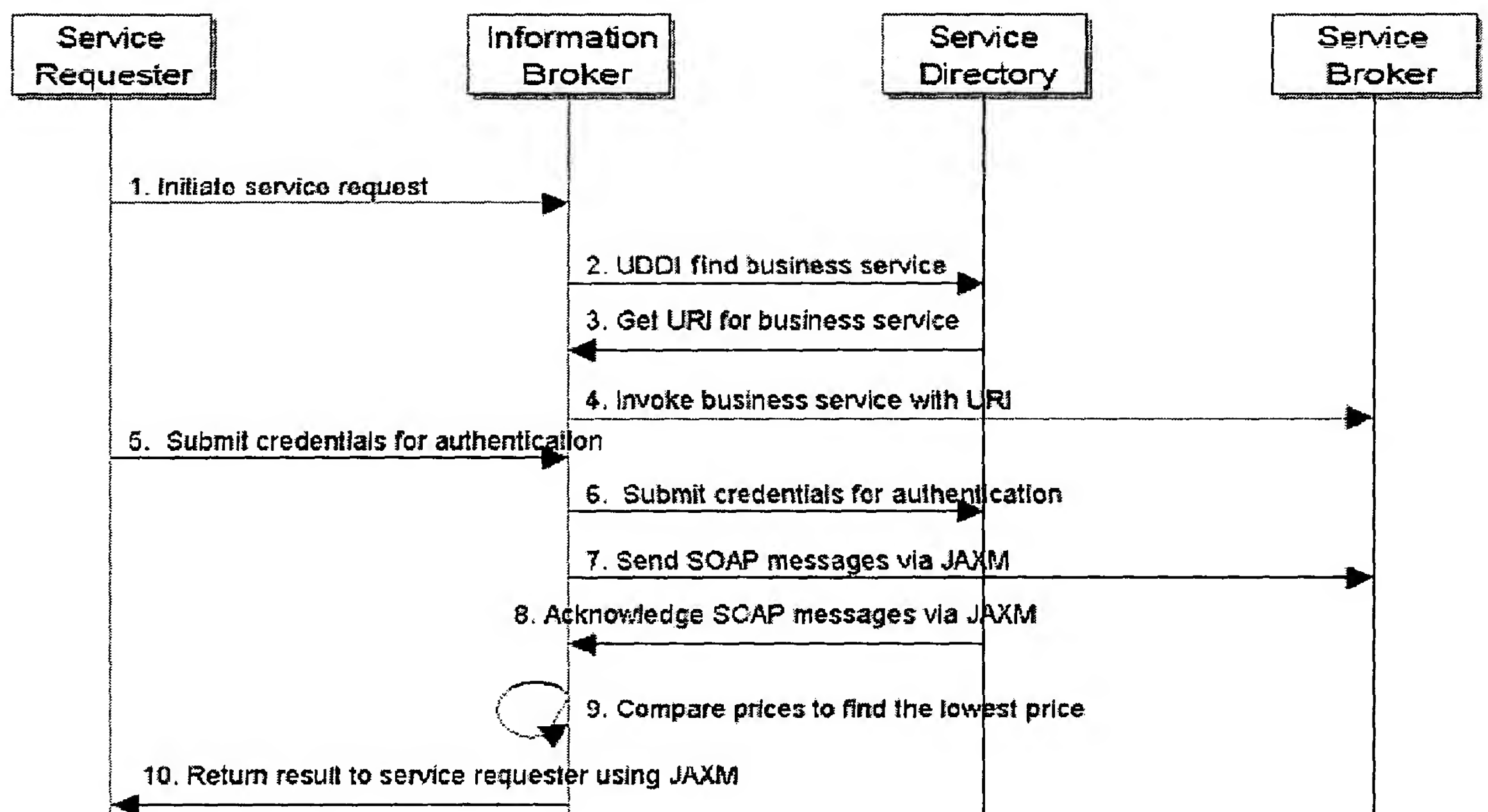


FIG. 90

Integration Patterns	When to Use	Benefits	Consideration
Application to Application	Point-to-point exchange	Tight integration	Limited scalability
Standard Build	Strong branding Strong urge to standardize	Reduce deployment effort Standardized service, faster deployment with no customization	Consensus on standard builds
Hub-Spoke Replication Federated Replication Multi-step Application Integration	Hub-spoke business model Intra-enterprise integration	Flexible workflow integration Reliable and consistent multi-step application integration	Inter-enterprise integration with many customization options
Data Exchange	Large number of partners to integrate with heterogeneous platforms & standards	Accommodating differences in standards/interfaces	Emerging standards and technology
Closed Process Integration Open Process Integration	Shared business processes Workflow-oriented services	Richer support for process integration Cohesive and tightly integrated services	Complexity for partners to agree and implement
Service Consolidation– Broker Integration Reverse Auction– Broker Integration	Single front-end for multiple Service Providers	Added values and Service-Provider neutral	Handling service failure of partners

**FIG. 91**



Integration Patterns	Typical Technology Used	Typical Standards Used	Examples
Application to Application	Customized adapters EDI translator	Proprietary XML variants	Ariba Commerce One
Standard Build	Proprietary	Proprietary	Hexagon
Hub-Spoke Replication Federated Replication Multi-step Application Integration	EAI solutions, such as Amtrix, Mercator, and TIBCO	JMS, SOAP-JMS binding	eXonomy
Data Exchange	XML Web Services	XML and SOAP, UDDI, WSDL	AIG Visa Commerce
Closed Process Integration Open Process Integration	EAI solutions or middleware, such as Sun ONE Integration Server EAI edition, XML Web Services technology	BPEL4WS	
Service Consolidation– Broker Integration Reverse Auction– Broker Integration	Hybrid of any integration technology	Hybrid of any integration standards	Yahoo! Digilogistics (obsolete) CFOWeb Vcheq (obsolete) Bumiputra Commerce Bank

FIG. 92

	Security Mechanism	Examples of Security Protection	Security Standards Specifications
Service Negotiation	Identity management  Access control and policy management Single Sign-on	Liberty-compliant Identity Server  Access control for XML messages Single Sign-on products	<i>Identity management</i> —Liberty 1.1, XML Key Management Specification (XKMS), WS-Federation <i>Entitlement</i> —SAML, XACML, WS-Authorization  <i>Policy</i> —WS-Policy <i>Others</i> —WS-Secure Conversation, WS-Trust, WS-Privacy
Service Discovery	Service Registry security	UDDI Service Registry security features Protection for WSDL documents	UDDI WSDL
Transaction Routing	Messaging security	Data encryption  Digital signature  Key management and managing credentials	XML Encryption (XML-ENC) XML Signature (XML-DSIG) WS-Security XKMS
Transport	Data transport security	128-bit SSL with HTTPS  Protocol security for FTP, SMTP, and so forth	HTTPS HTTPR IPSec
Internet	Network connectivity security	Leased line or router-level encryption Virtual Private Network (VPN) gateways	
Platform	Operating system security  Penetration testing  Key exchanges between hosts	Solaris OE™ hardening  Linux Operating System (OS) hardening  Windows OS hardening  Professional Penetration Testing	

FIG. 93

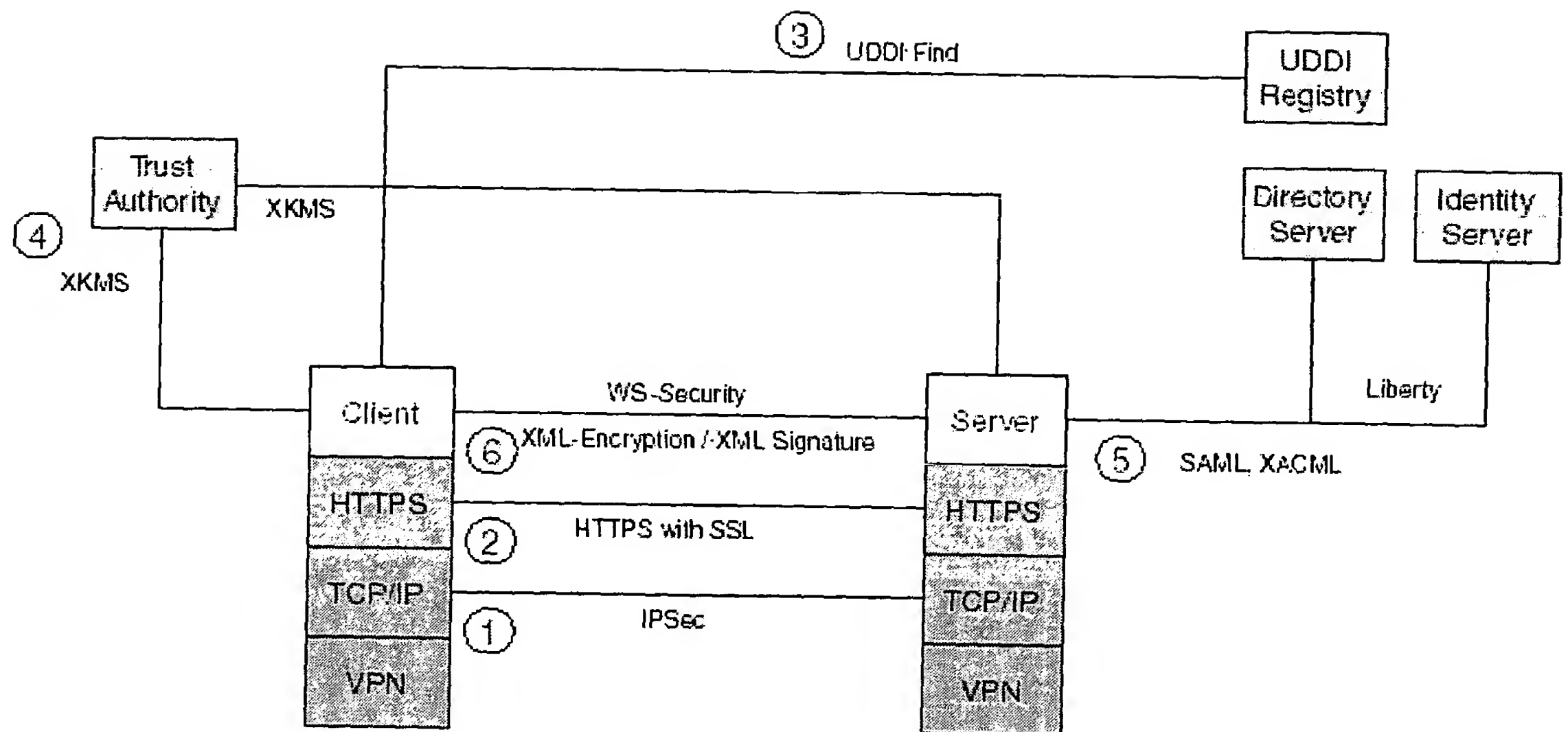


FIG. 94

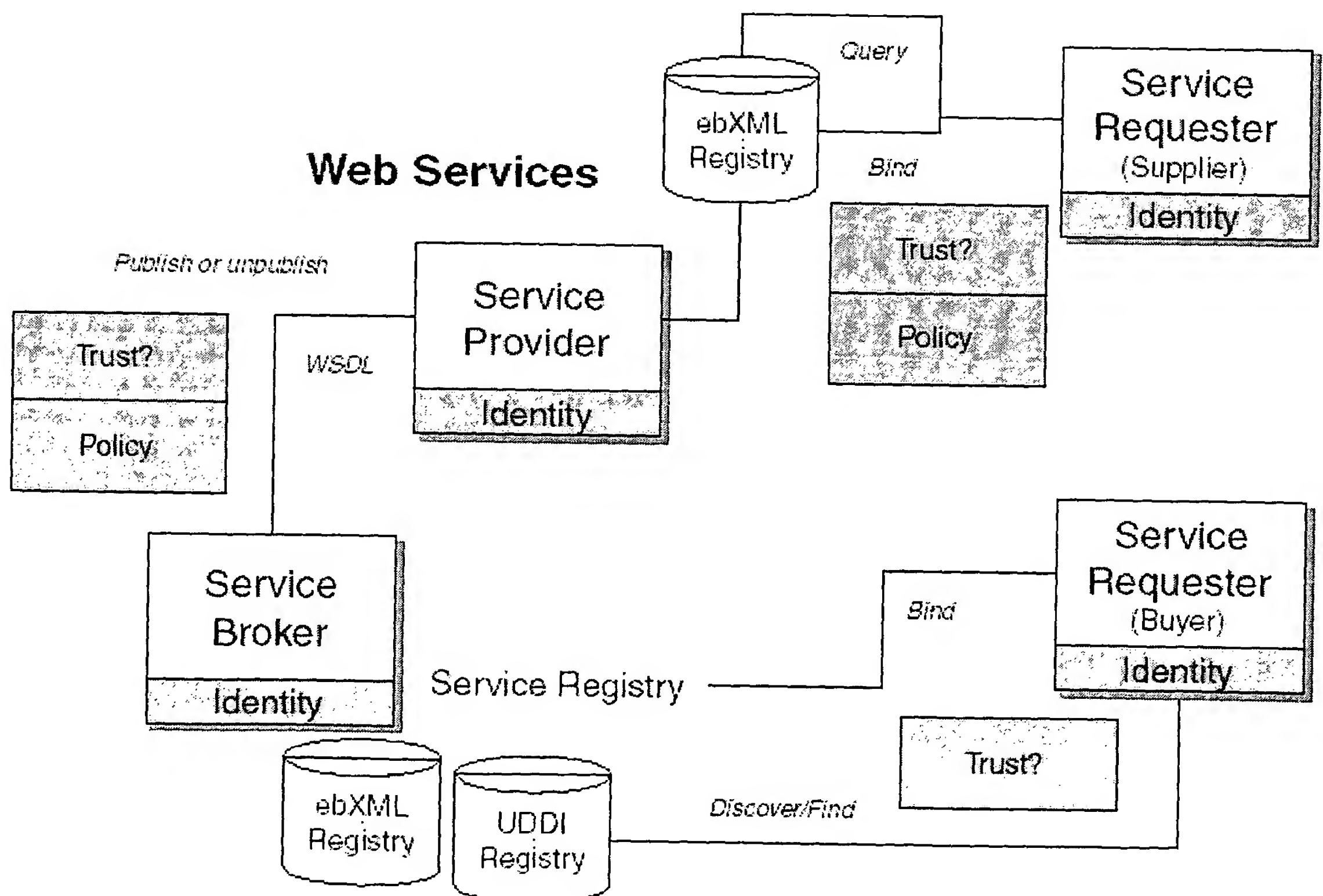


FIG. 95

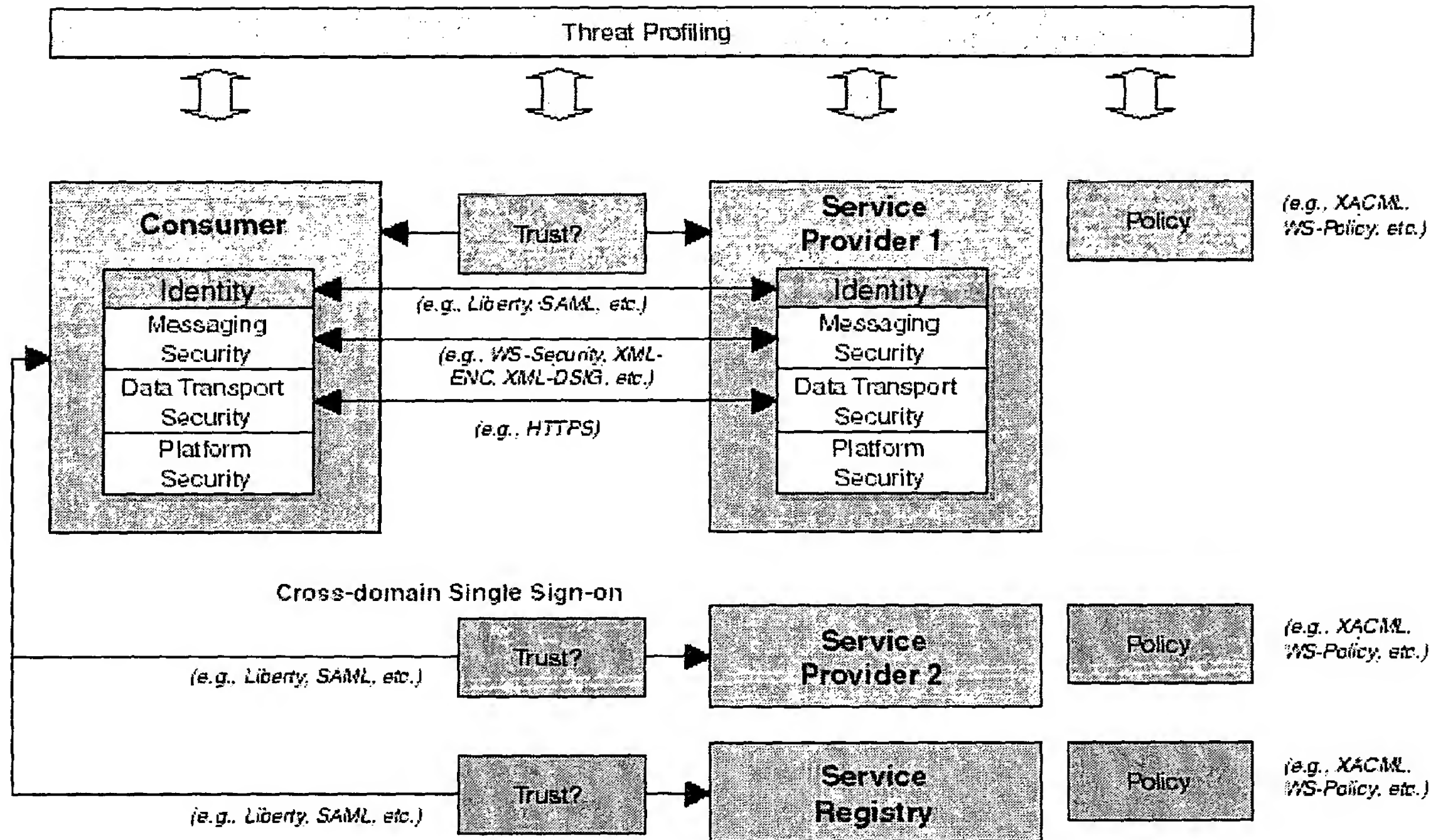


FIG. 96

	Security Technology or Standards	Security Requirements
<b>Trust Domains</b>		
Key management	XKMS Host security hardening	Authentication Confidentiality Traceability Non-repudiation
Authentication	Single Sign-on with SAML and Directory Server	Authentication Entitlement Traceability Availability
Transactional security	XML Encryption, XML-DSIG XACML WS-Security Client and host security hardening	Entitlement Confidentiality Availability Data integrity Non-repudiation
<b>Threat Profiling</b>		
Web Services objects	Security hardening for UDDI configuration files and WSDLs	Data integrity Availability
Hacker attack	Profiling of transaction loading/capacity to support availability and scalability Client and host security hardening Virus protection for hosts Intrusion detection testing Patch management for software platform (for example, buffer overflow)	Availability Confidentiality Traceability Authentication Entitlement Non-repudiation

**FIG. 97**



Acquire a WSDL document and sniff for a copy of SOAP message from internal network

```

<message name="transferFundRequest">
  <part name="account1" type="xsd:string"/>
  <part name="account2" type="xsd:string"/>
</message>
<message name="transferFundResponse">
  <part name="Result" type="xsd:float"/>
</message>

```

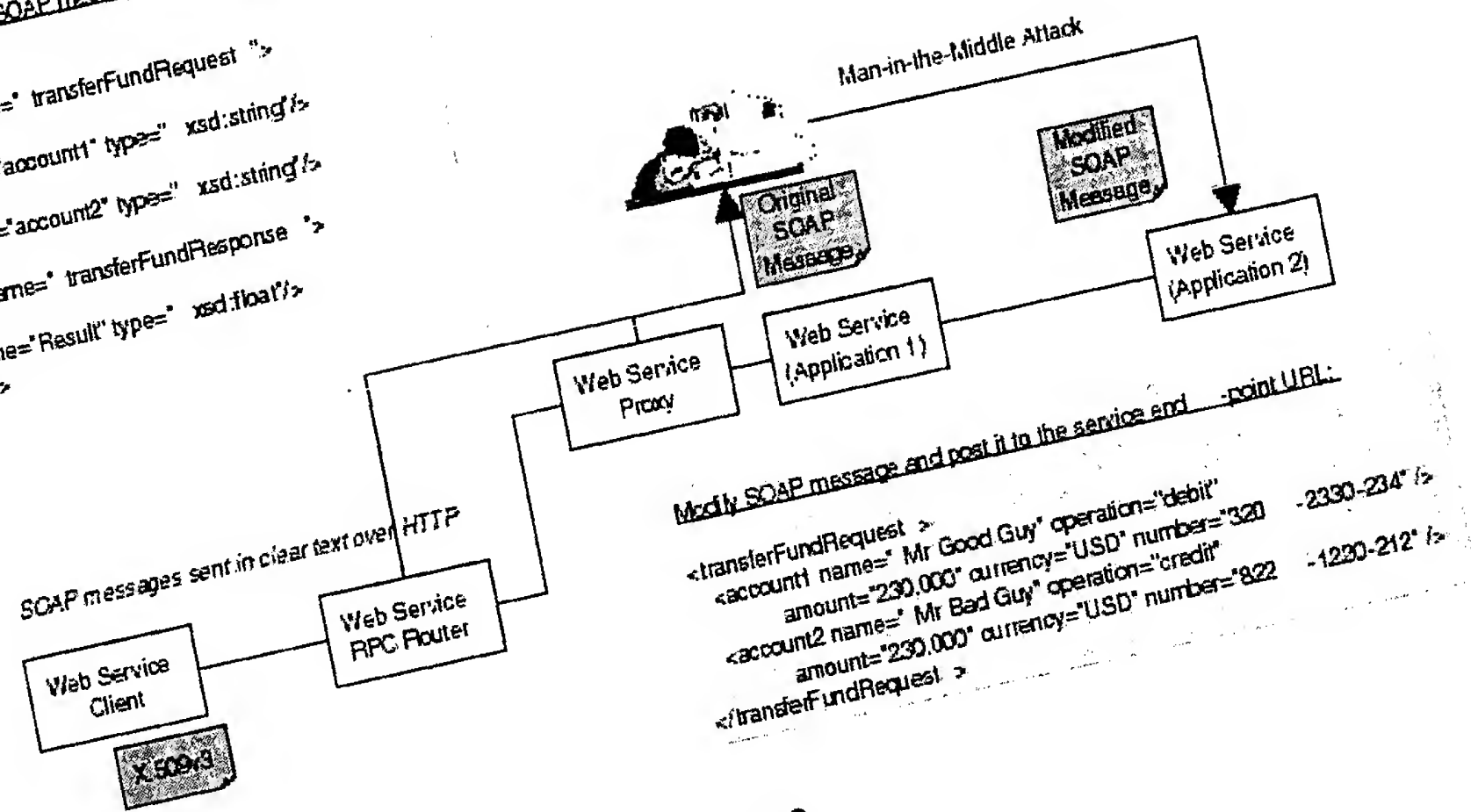


FIG. 98

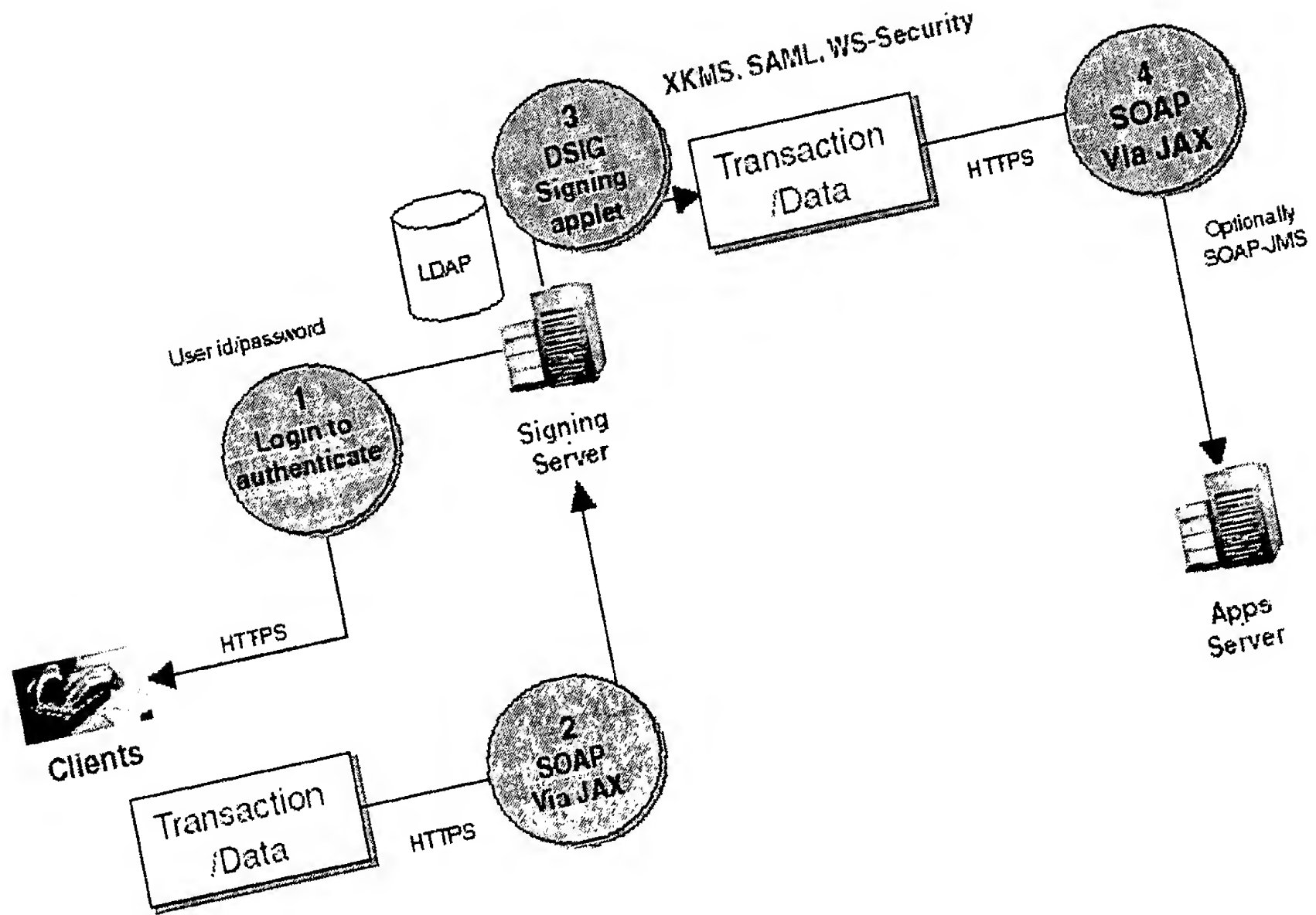


FIG. 99

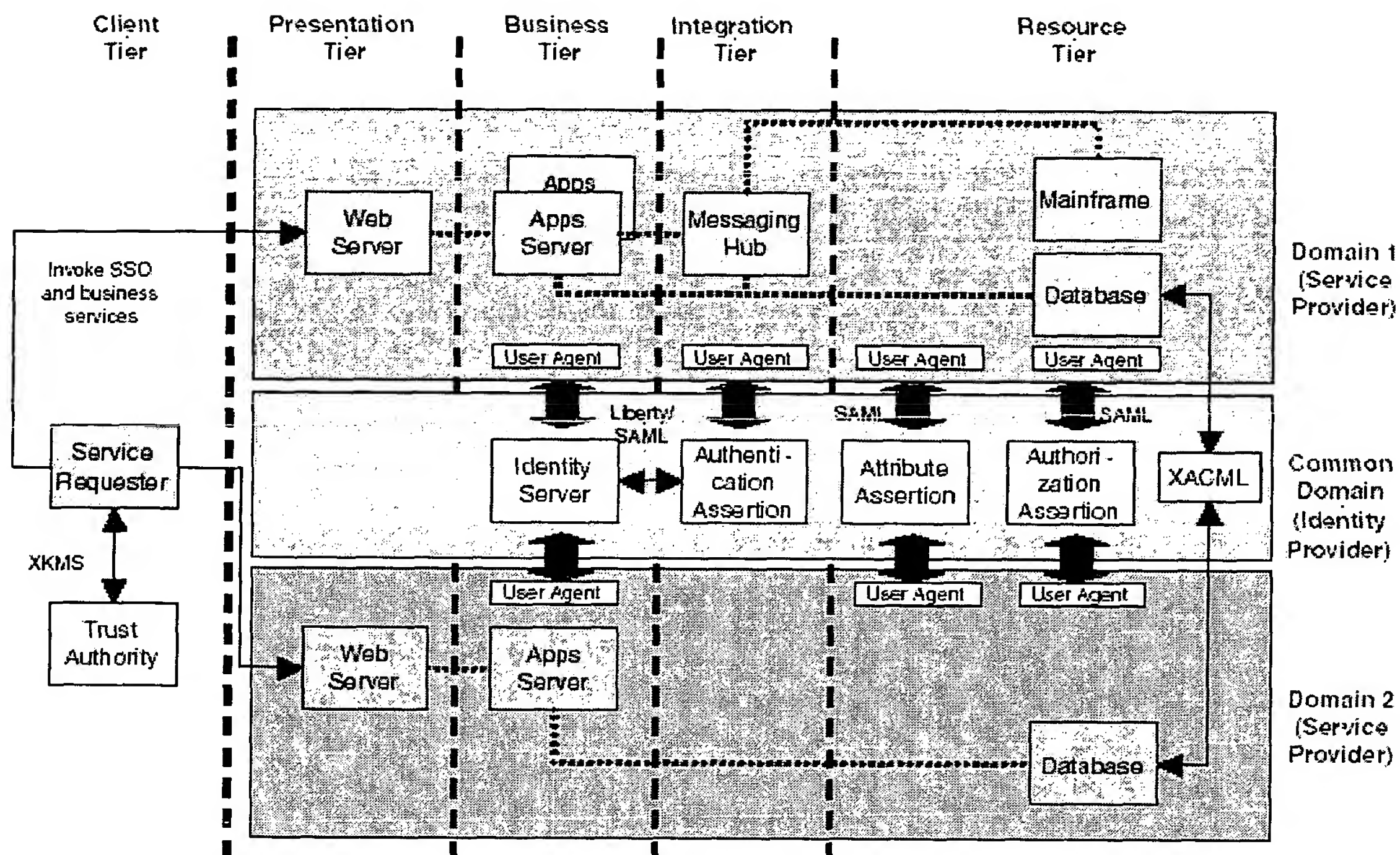


FIG. 100

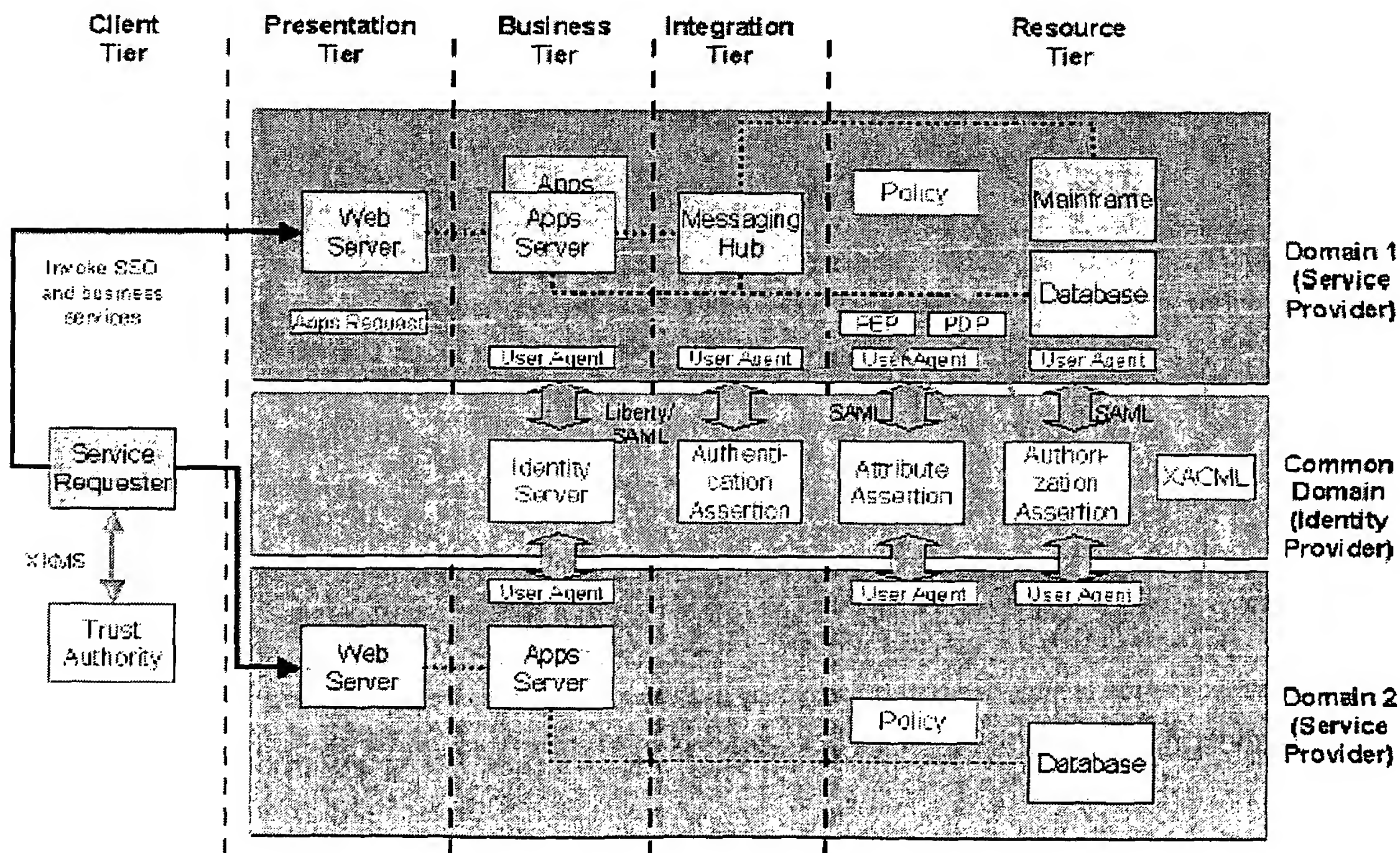


FIG. 101

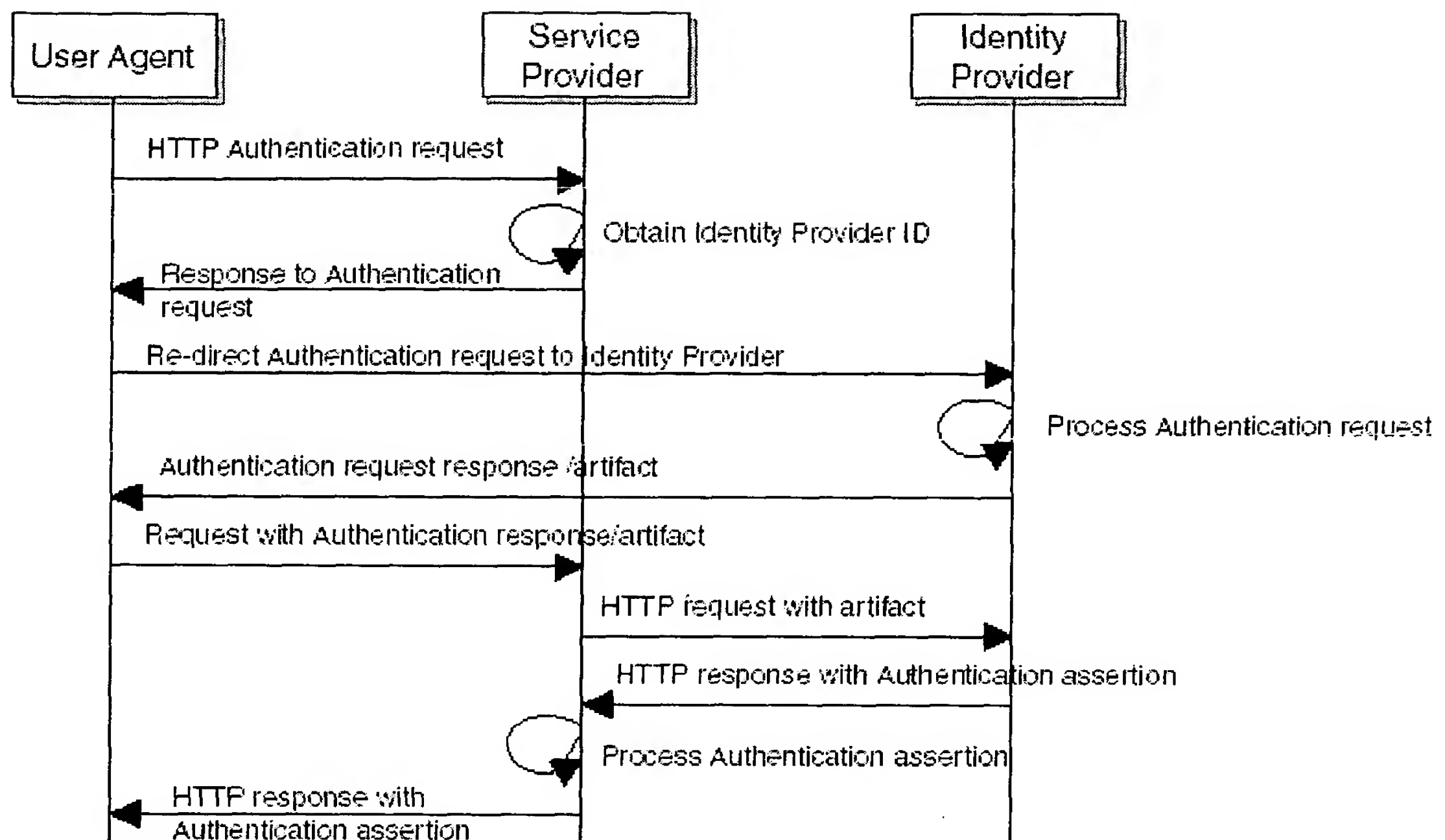


FIG. 102

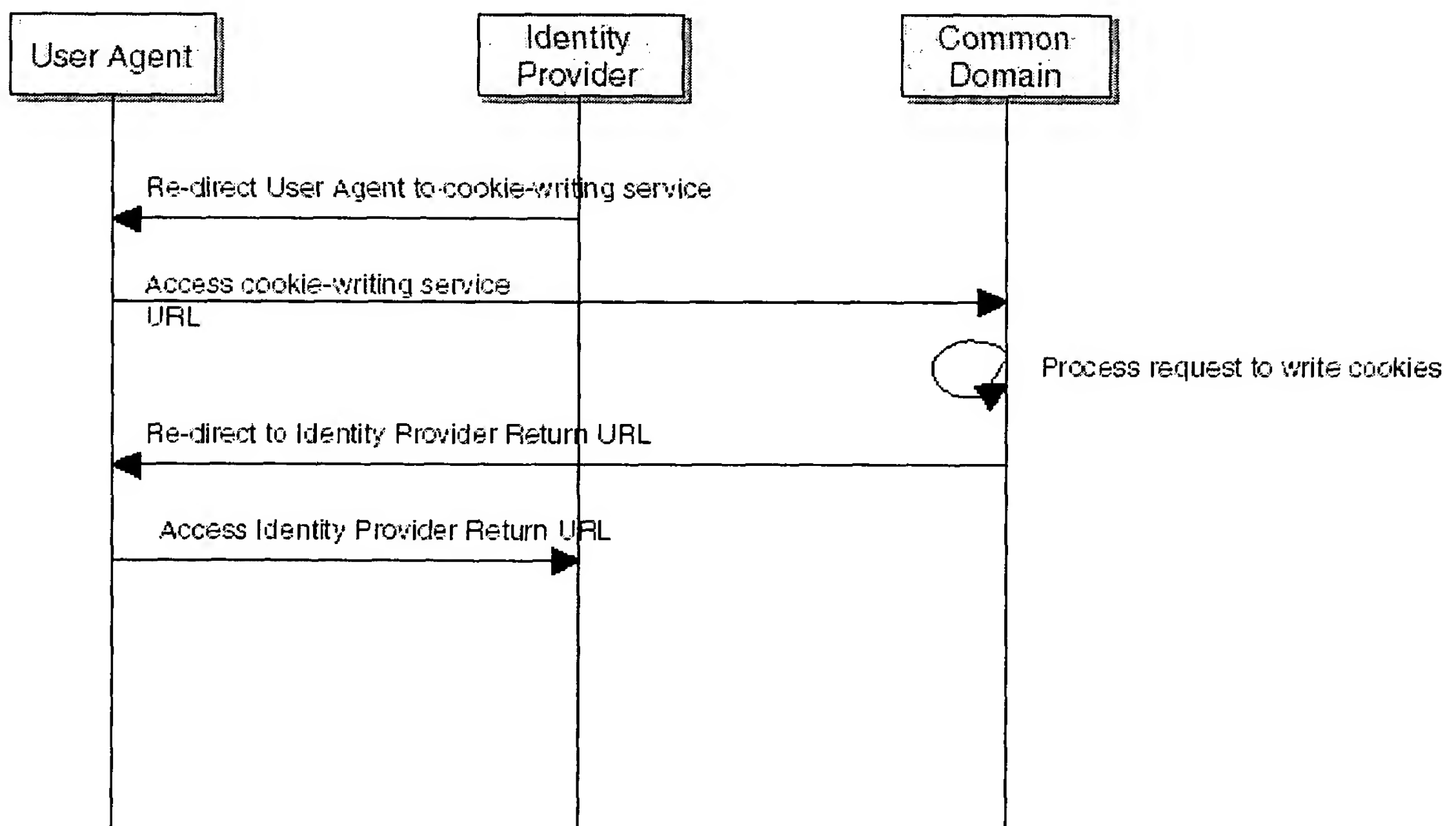


FIG. 103



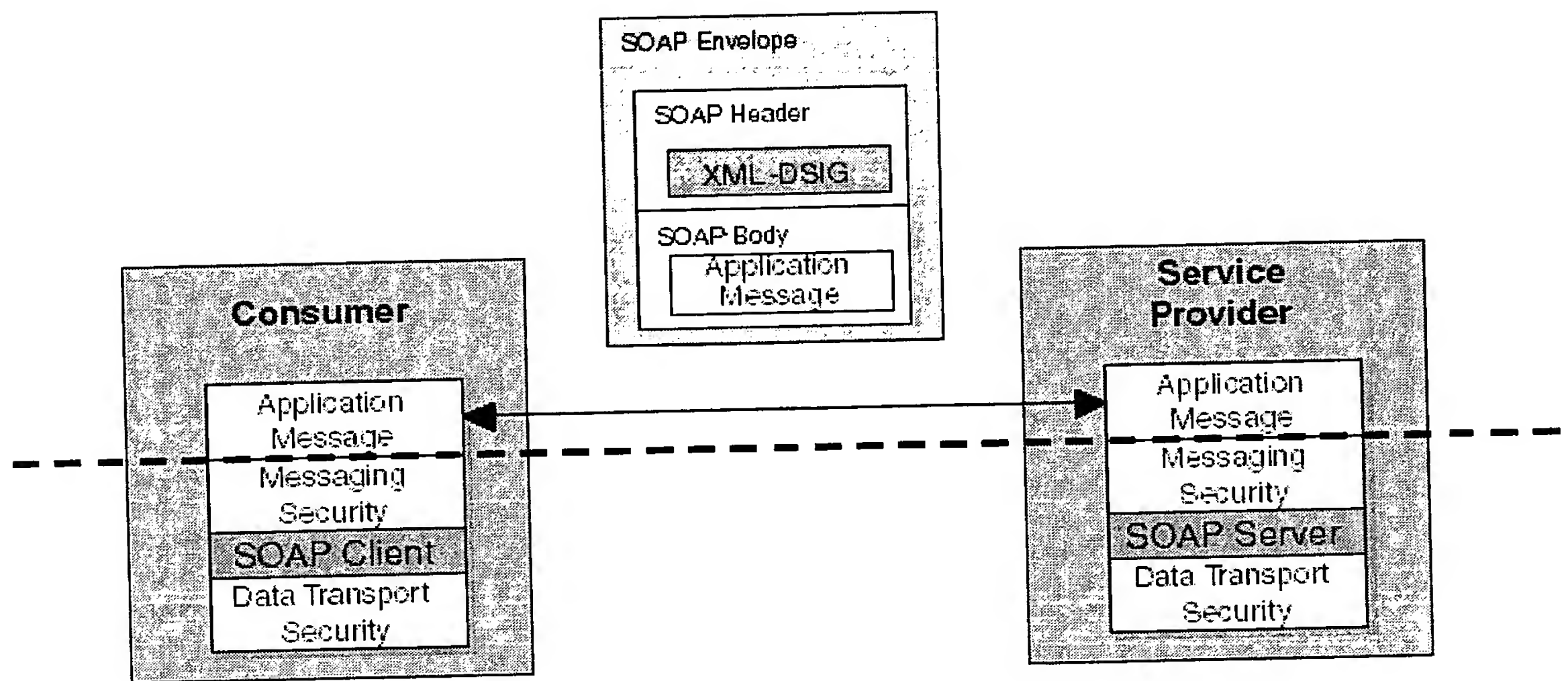


FIG. 104

Web Services Objects		
Objects	Location	Remarks
<b>Web Container</b>		In this example, this is Apache Tomcat 4.x.
User access control list	D:\Dev\WSDP\conf\tomcat-users.xml	This file contains the user names, user passwords, and roles that are allowed to access and execute resources under the Web Container.
Server configuration file	D:\Dev\WSDP\conf\server.xml	This file contains the server configuration (for example, port number) for running the Tomcat server.
<b>Log Files</b>		
Web Container log files	D:\Dev\WSDP\logs	In this example, Tomcat log files are used. This directory contains log files for Tomcat server (Catalina.out), server administration log (localhost_admin_log*.log and access_log*.log and services_log*.log), as well as Service Registry log (xindice.log).
Developer tool log files	D:\Dev\WSDP\logs\jwsdp_log*.log	In this example, Java Web Services Developer Pack's log files are shown.
Service Registry update activity log file	D:\Dev\WSDP\tools\xindice\logs\xindice.log	In this example, the Xindice database activity log file is used.
<b>Message Provider</b>		
ebXML message provider administration logs	D:\Dev\WSDP\work\Services Engines\jaxm-provider\ebxml	There are four subdirectories that contain the messages received, sent, to be dispatched, and to be sent. This denotes the physical location where the JAXM message provider will send or receive the messages with the reliable message delivery capability.

FIG. 105A



Web Services Objects	Location	Remarks
SOAP Remote Provider message provider administration logs	D:\Dev\WSDP\work\Services Engines\jaxm-provider\soaprp	There are four subdirectories that contain the messages received, sent, to be dispatched, and to be sent. This denotes the physical location where the SOAP remote message provider will send or receive the messages with the reliable message delivery capability
<b>Service Registry</b>		In Java Web Services Developer Pack, UDDI Service Registry is implemented using Xindice object database.
Service Registry files	D:\Dev\WSDP\tools\xindice\db	This file location contains the subdirectory 'system' for the object database system files and security information, and the subdirectory 'uddi' for the actual UDDI data store.
WSDL documents	N/A	In this demo environment, the WSDL documents are generated dynamically and do not store in the Service Registry.

FIG. 105B

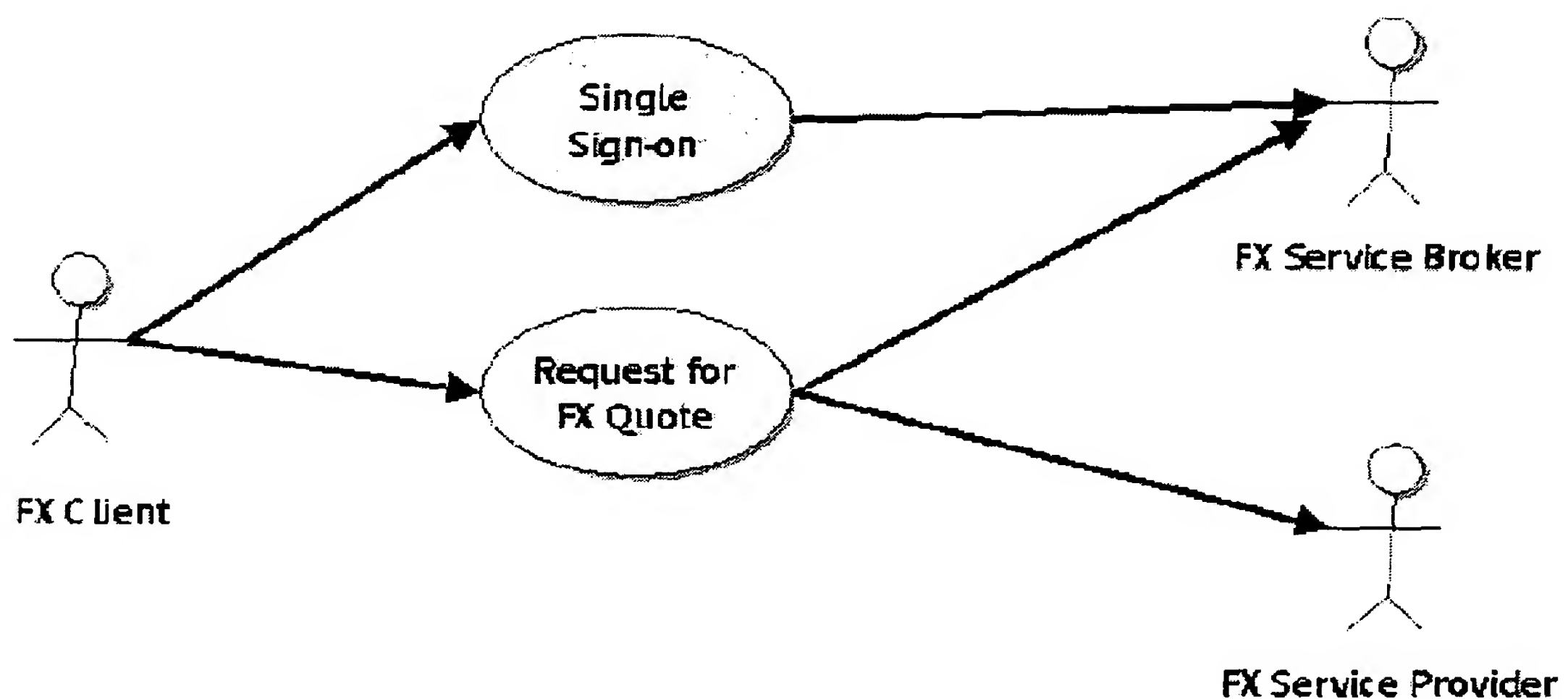


FIG. 106

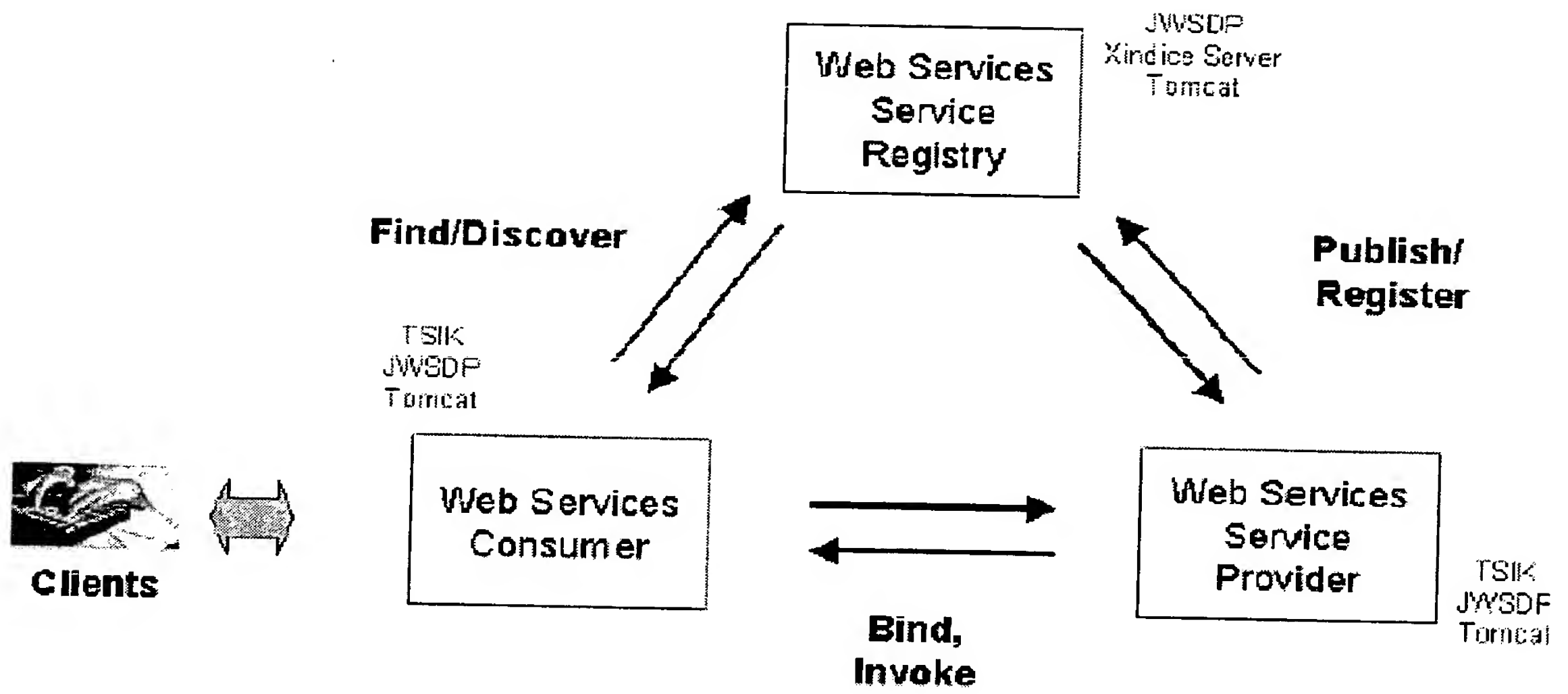


FIG. 107

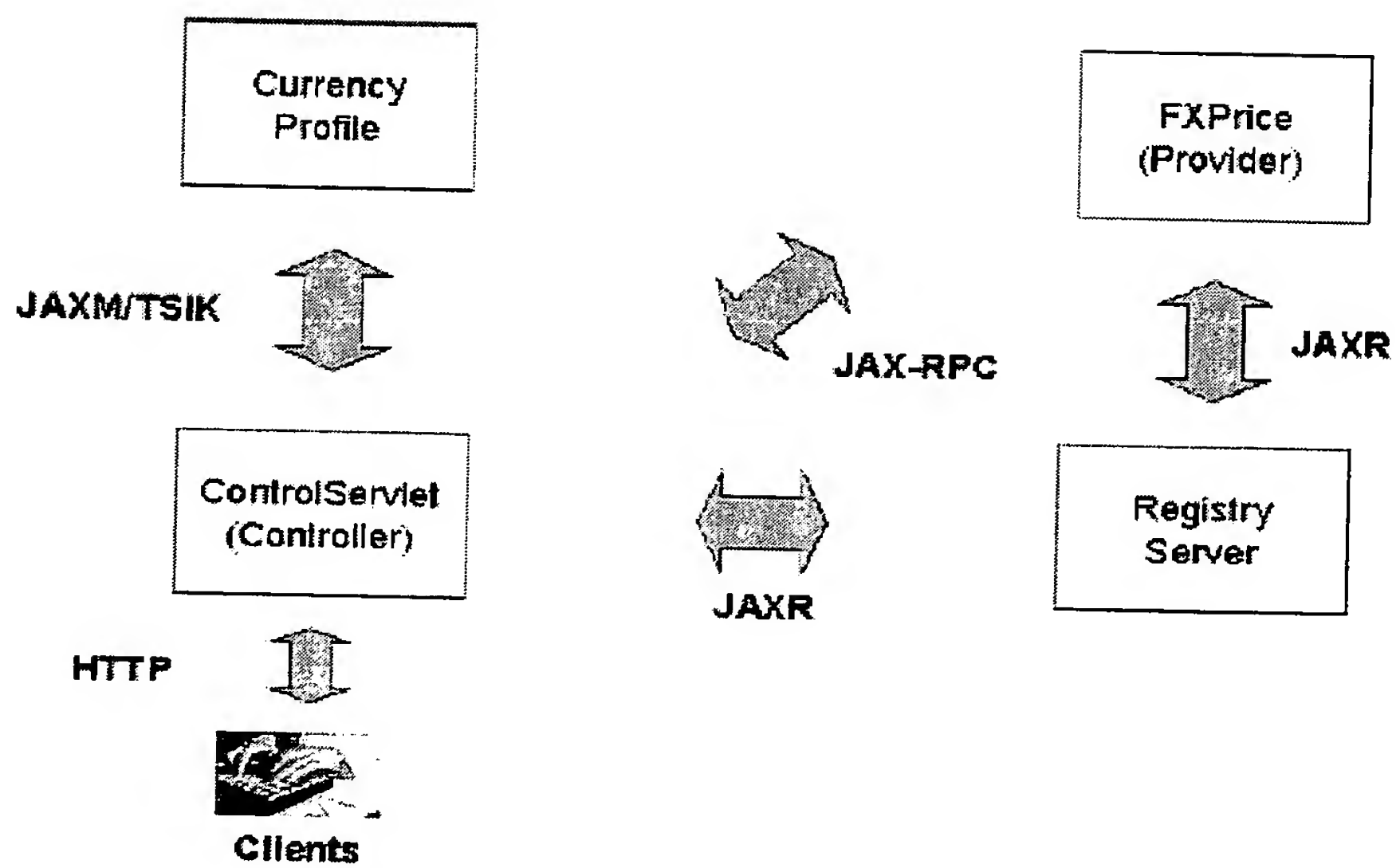


FIG. 108

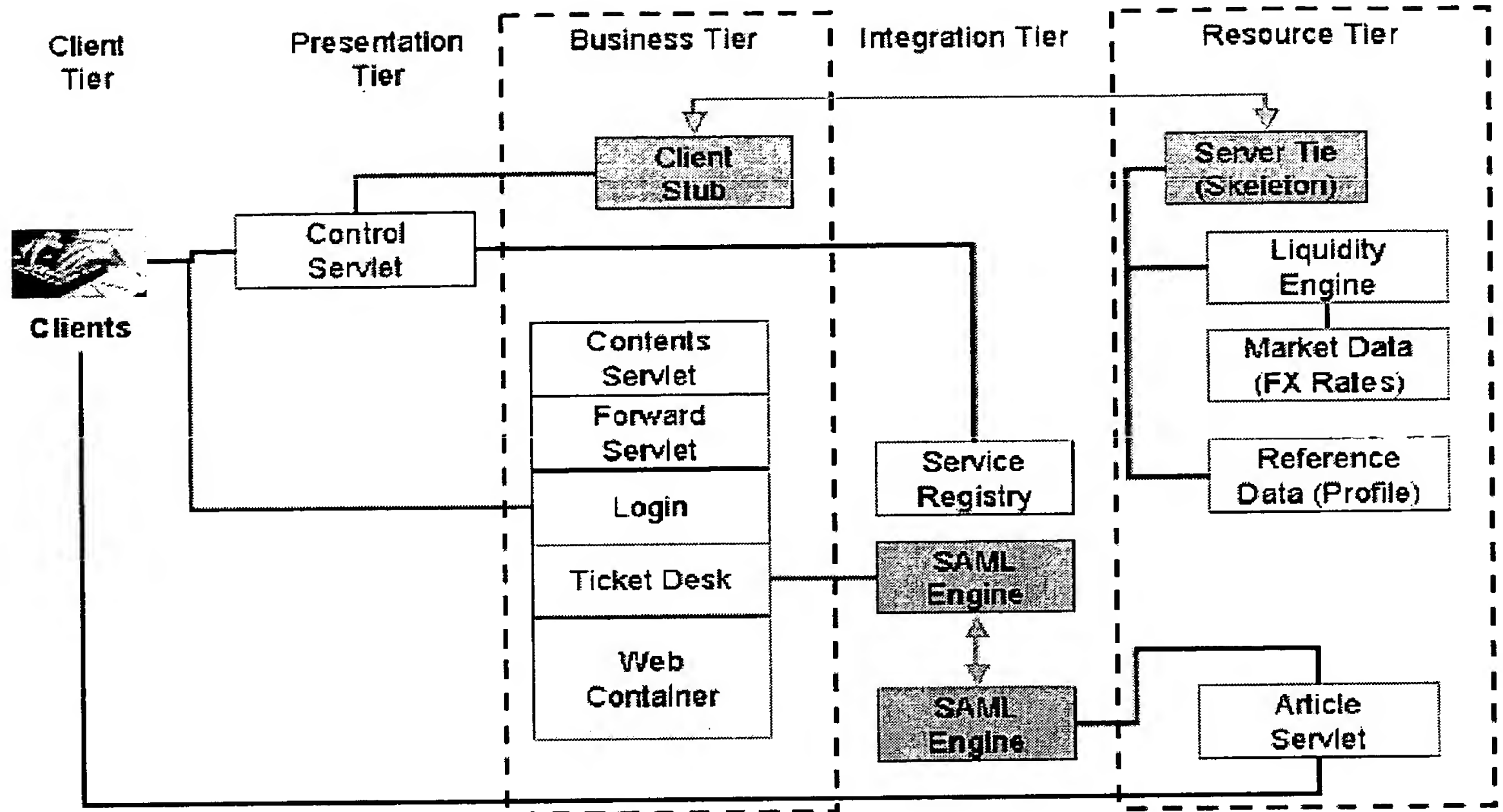


FIG. 109

	Tiers				
	Client	Presentation	Business	Integration	Resource
<b>Application Platform Layer</b>	User id and password are used for authentication.	Control Servlet uses HTML and JSP for presentation and inquiry.  JSPs can be cached to enhance performance.	Java beans are used to implement some of the business logic.  The remote FX Quote Service is a black box, accessible via JAX-RPC.	N/A	N/A
<b>Virtual Platform Layer</b>	HTTP  HTTPS with SSL can be used for better security.	HTTP  HTTPS with SSL can be used for better security.	JAXM-TSIK Message Provider provides secure messaging transport for SOAP messages over HTTP.	JAX-RPC, JAXM are used to integrate different remote services.	JAXR is used to access the Service Registry.
<b>Upper Platform Layer</b>	In the future, 128-bit SSL can be used for better security.	HTTP load balancing can be used for better scalability.	N/A	In the future, server clustering can be used for availability.	In the future, server clustering can be used for availability.
<b>Lower Platform Layer</b>	Basic Operating System security is provided with id and password.	N/A	N/A	N/A	N/A
<b>Hardware Platform Layer</b>	SSL accelerator can be added in the future for faster performance when using HTTPS.	Reliability and securability can be enhanced in the future with server hardening, firewall configuration, and hardware clustering.	Reliability and securability can be enhanced in the future with server hardening, firewall configuration, and hardware clustering.	N/A	Reliability and securability can be enhanced in the future with server hardening, firewall configuration, and hardware clustering.

FIG. 110

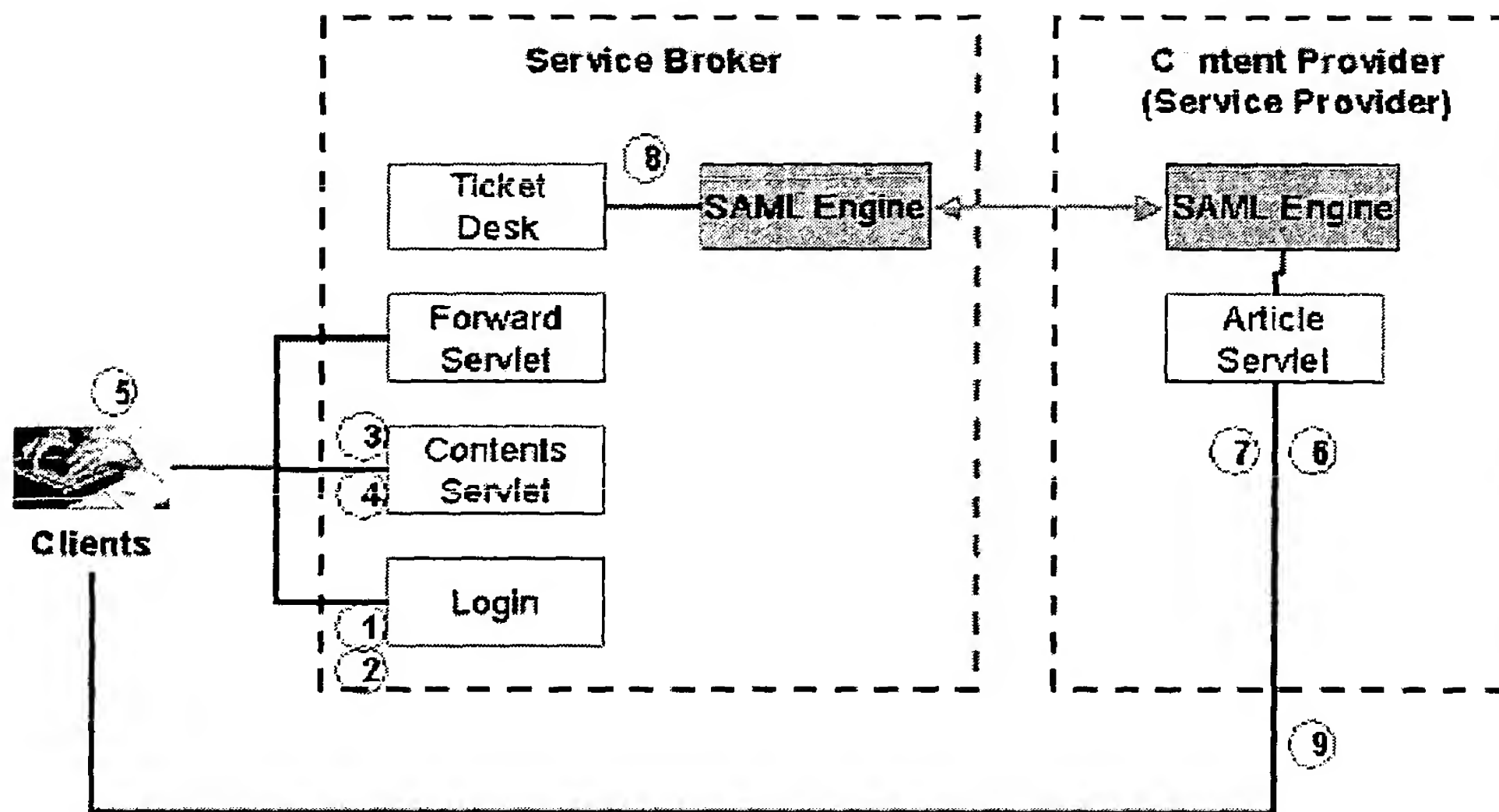


FIG. 111

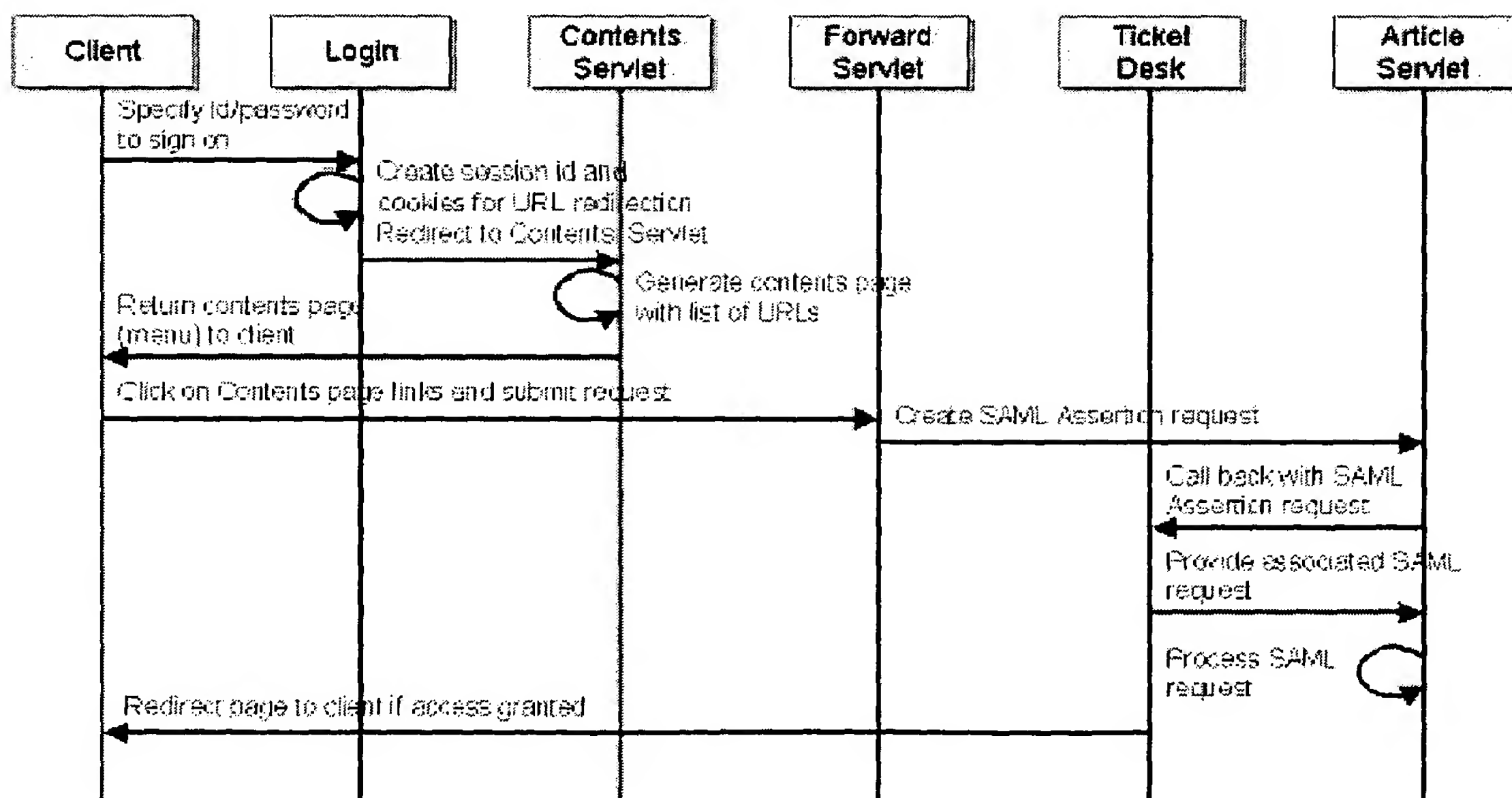


FIG. 112



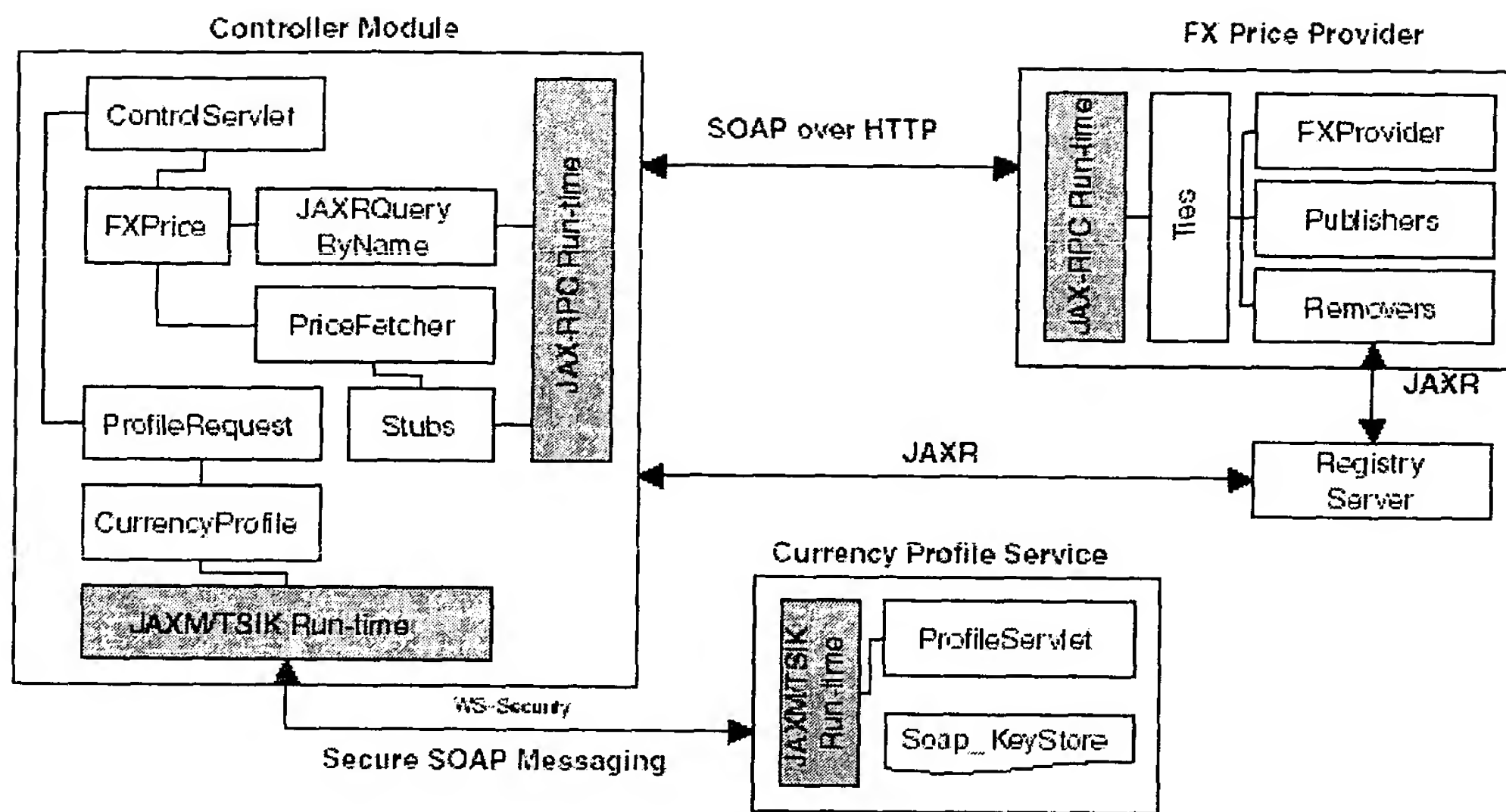


FIG. 113

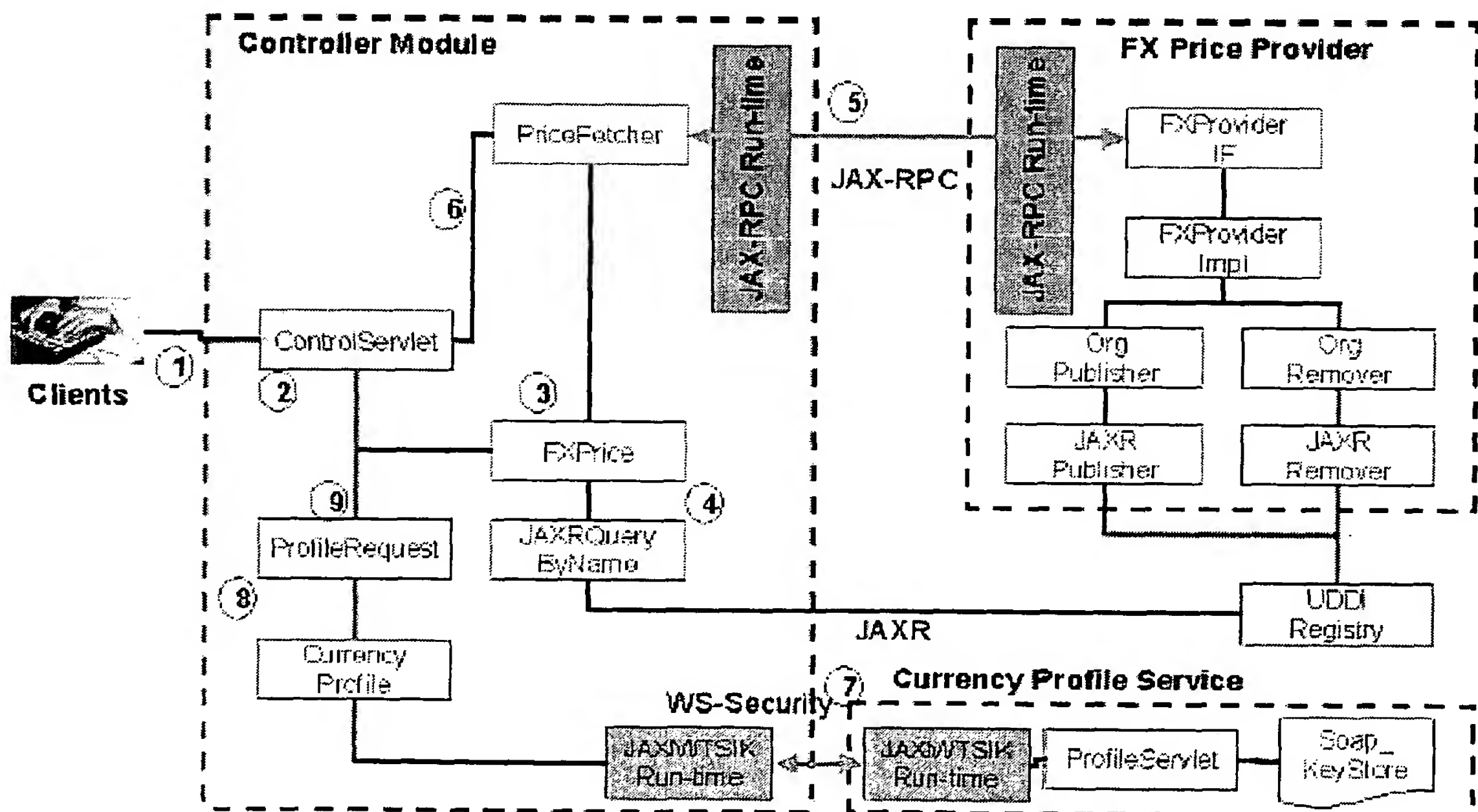


FIG. 114

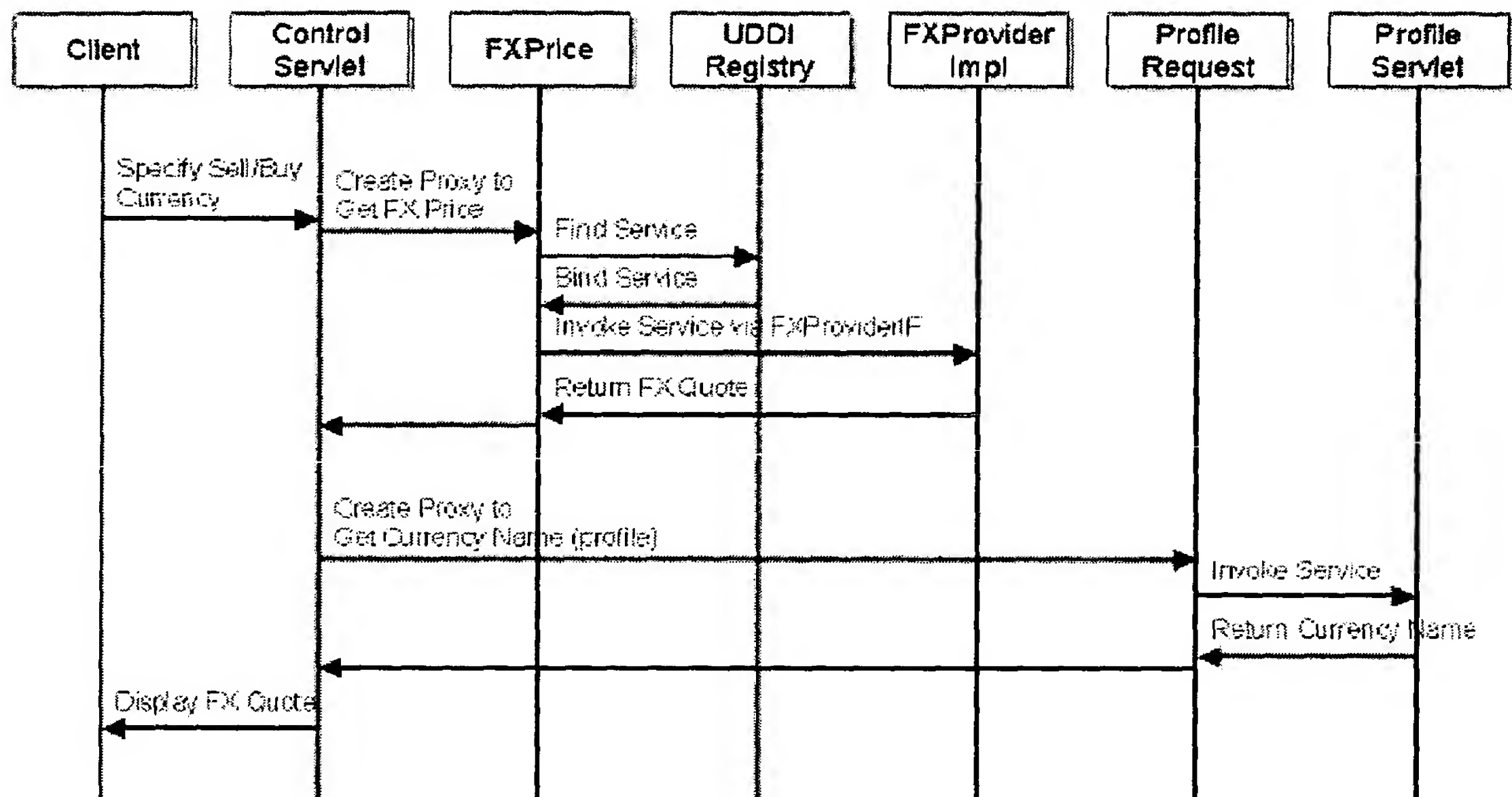


FIG. 115

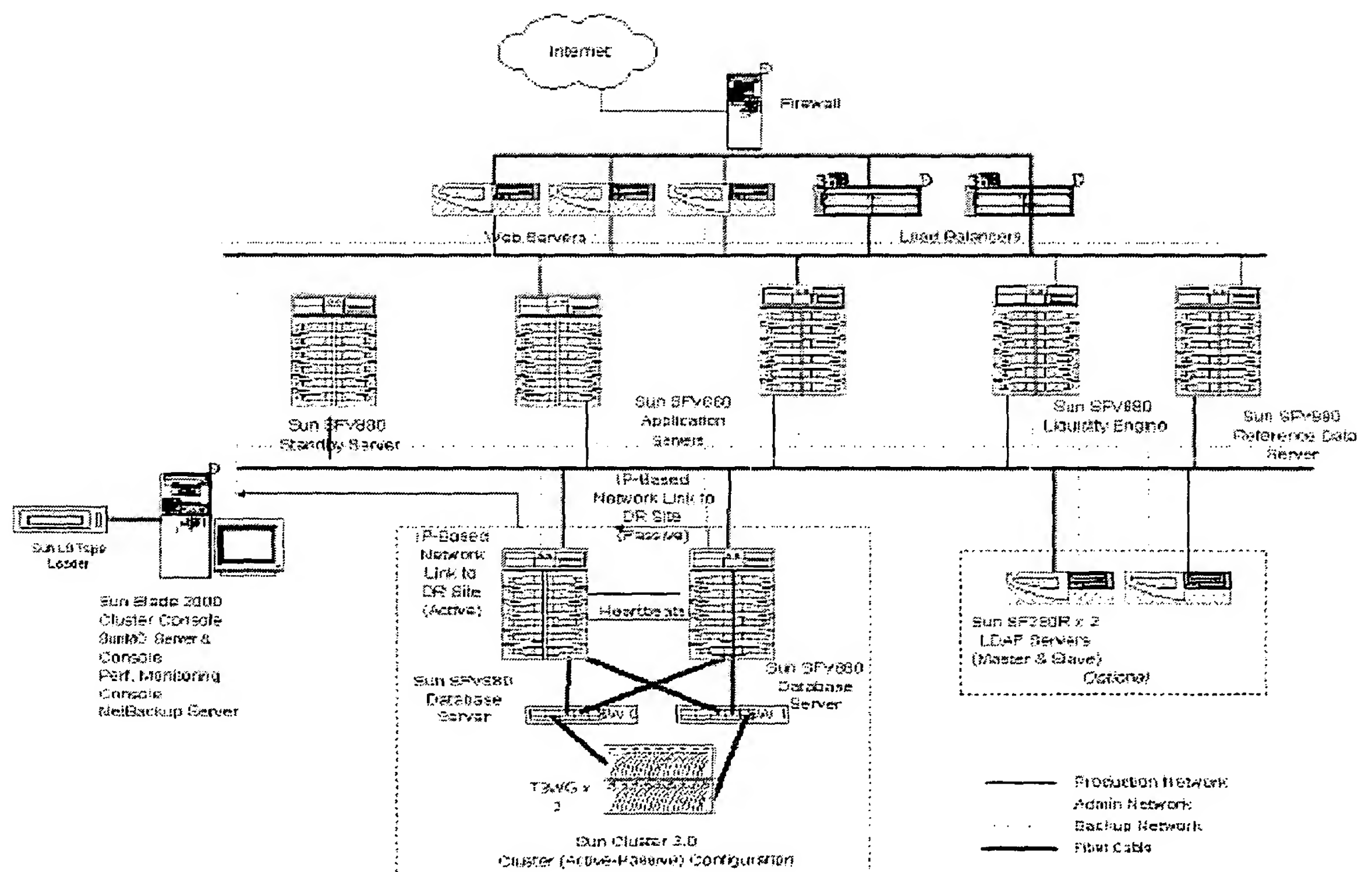


FIG. 116

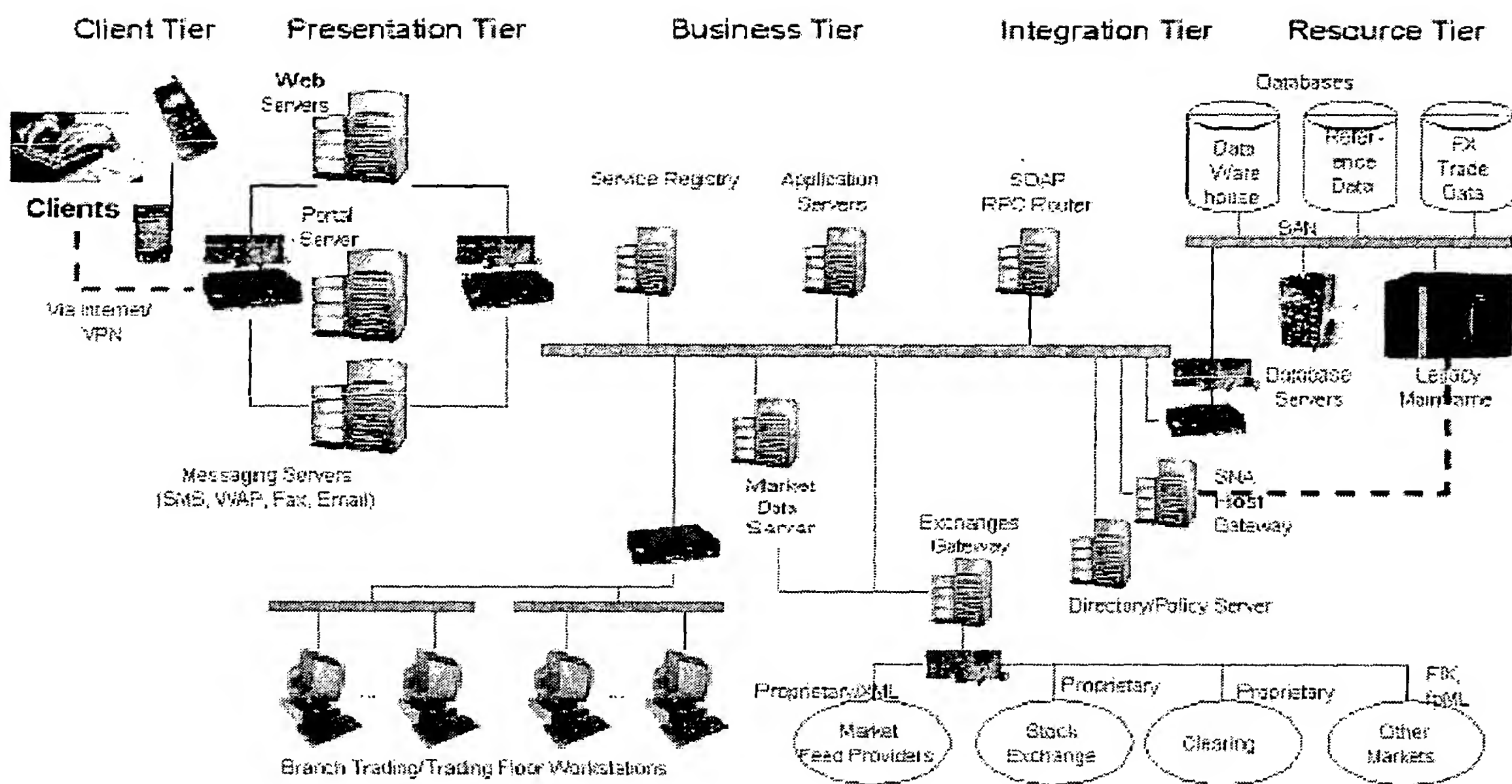
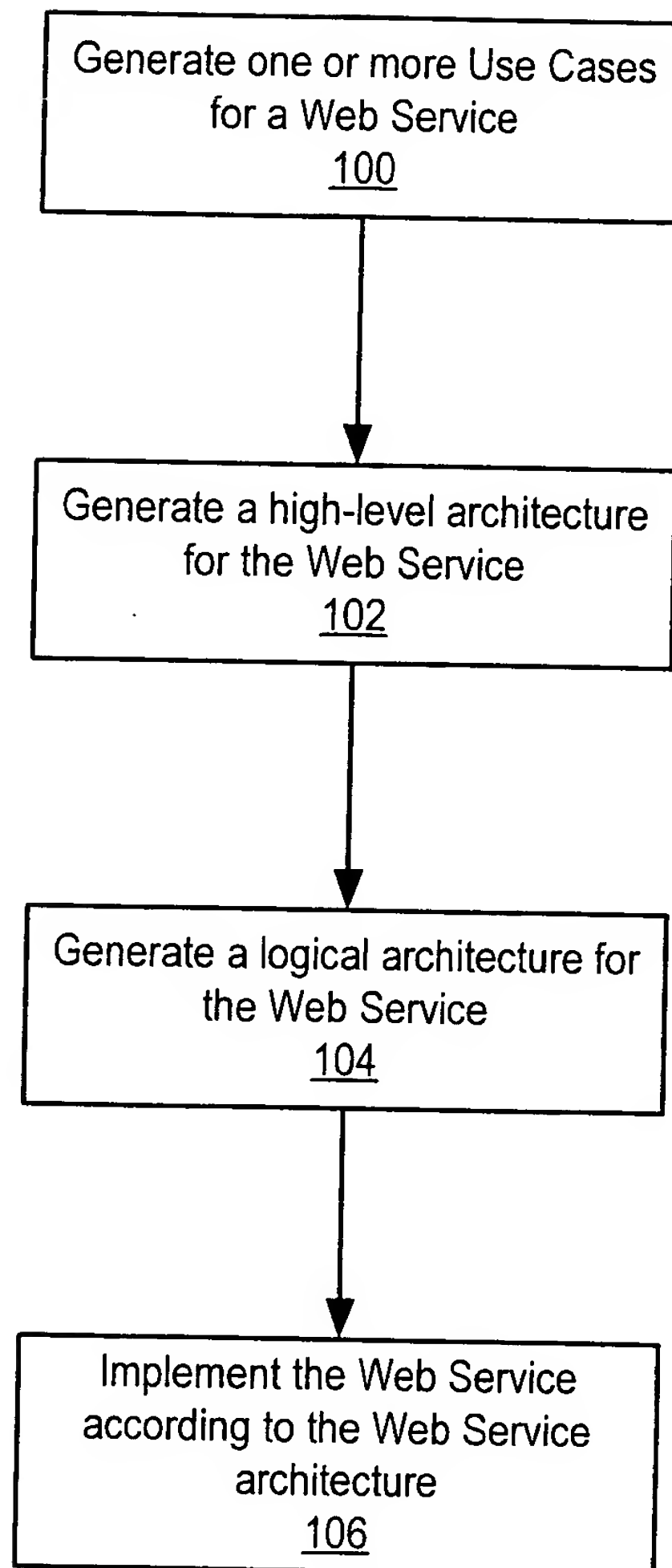


FIG. 117



*FIG. 118*

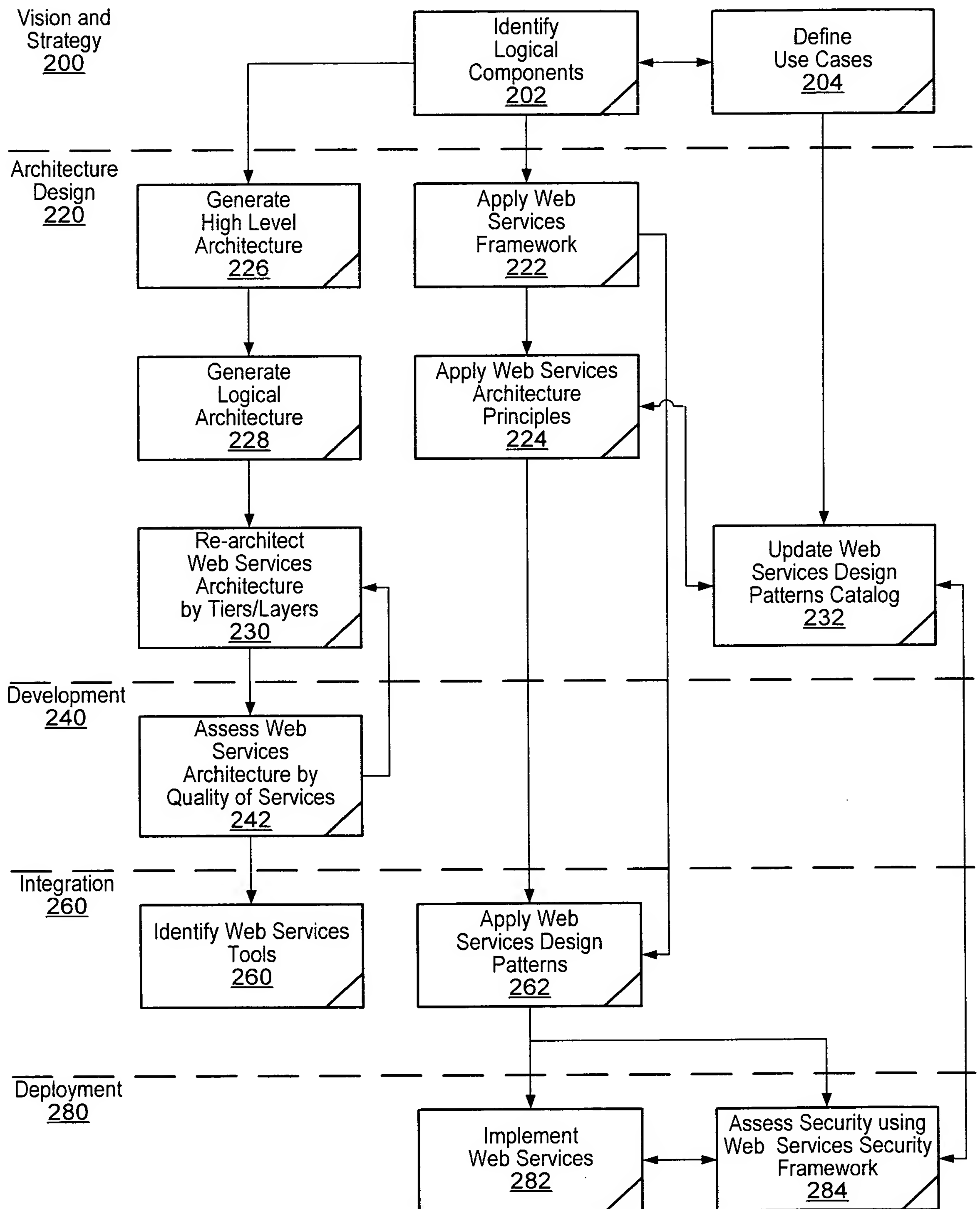
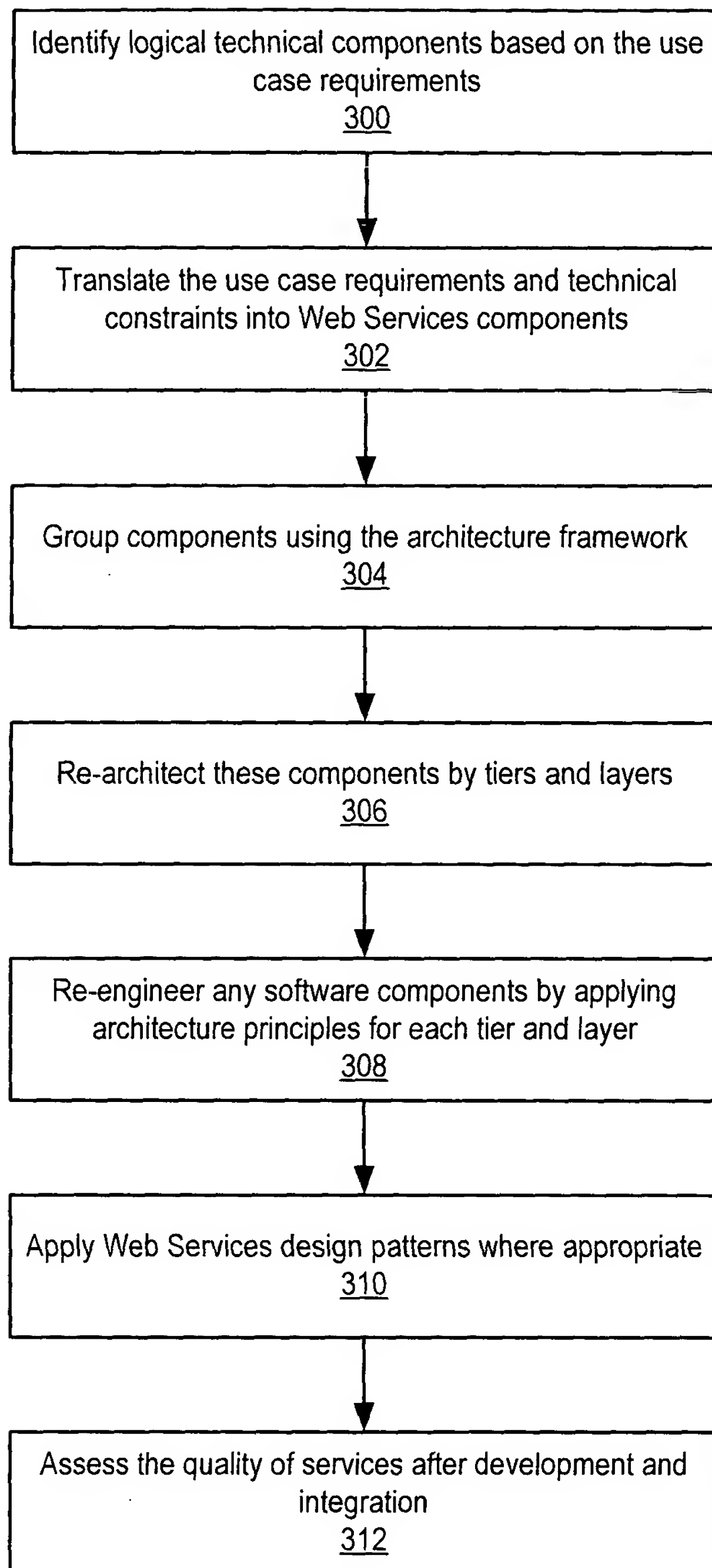


FIG. 119





*FIG. 120*

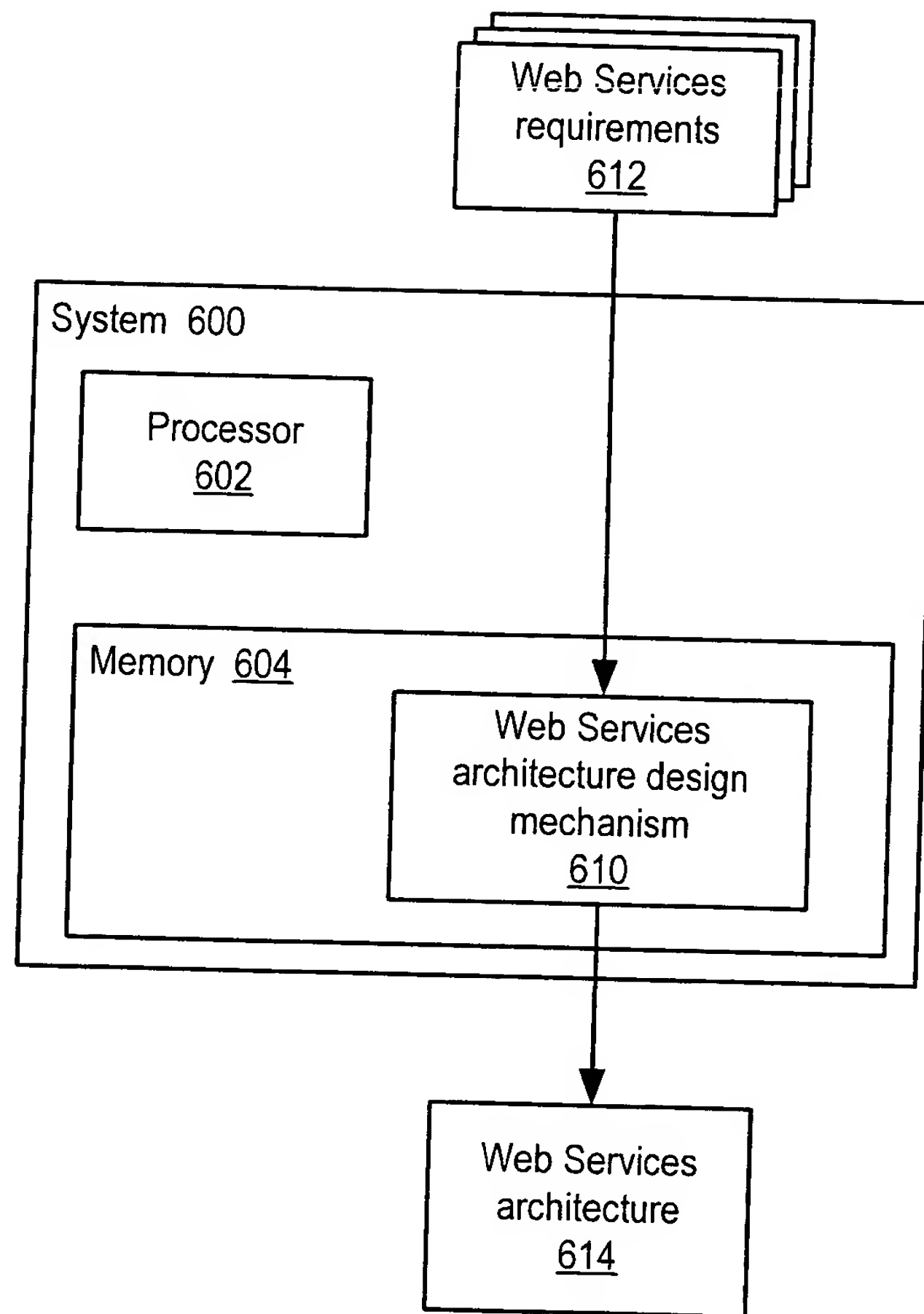


FIG. 121

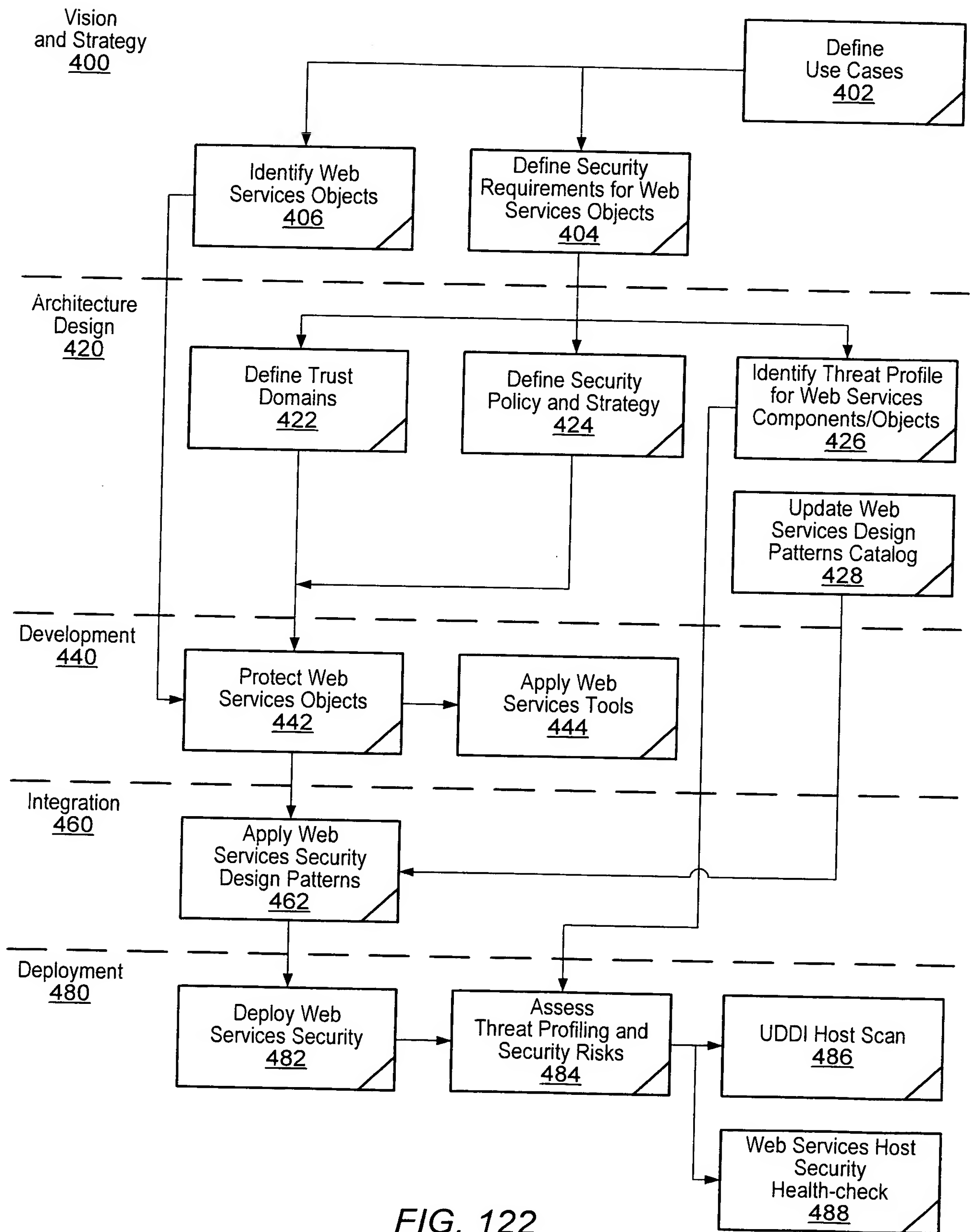
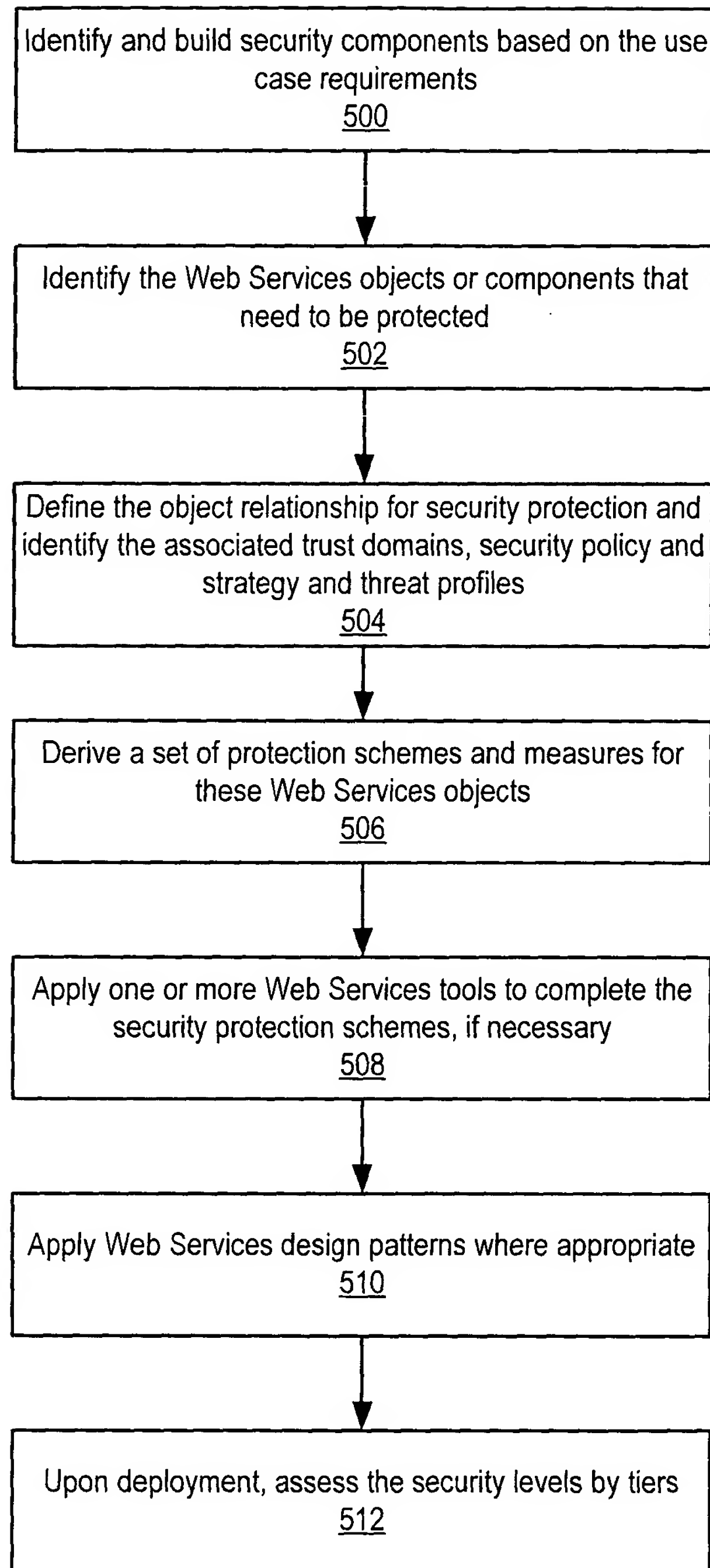


FIG. 122



*FIG. 123*